Windows 10, version 1809~~1703~~ basic level Windows diagnostic events and fields

- Windows 10, version 1809~~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~~Windows~~ Store. When the level is set to Basic, it also includes the Security level information.

The Basic level 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:

- Windows 10, version 1803~~1809~~ basic diagnostic events and fields
- Windows 10, version 1709~~1803~~ basic diagnostic events and fields
- Windows 10, version 1703~~1709~~ basic diagnostic events and fields
- Manage connections from Windows operating system components to Microsoft services
- Configure Windows diagnostic data in your organization

AppLocker events

Microsoft.Windows.Security.AppLockerCSP.ActivityStoppedAutomatically

Automatically closed activity for start/stop operations that aren't explicitly closed.

Microsoft.Windows.Security.AppLockerCSP.AddParams

Parameters passed to Add function of the AppLockerCSP Node.

The following fields are available:

- **child** The child URI of the node to add.
- **uri** URI of the node relative to %SYSTEM32%/AppLocker.

Microsoft.Windows.Security.AppLockerCSP.AddStart

Start of "Add" Operation for the AppLockerCSP Node.

Microsoft.Windows.Security.AppLockerCSP.AddStop

End of "Add" Operation for AppLockerCSP Node.

The following fields are available:

- **hr** The HRESULT returned by Add function in AppLockerCSP.

Microsoft.Windows.Security.AppLockerCSP.CAppLockerCSP::Rollback

Result of the 'Rollback' operation in AppLockerCSP.

The following fields are available:

- **oldId** Previous id for the CSP transaction.
- **txId** Current id for the CSP transaction.

Microsoft.Windows.Security.AppLockerCSP.ClearParams

Parameters passed to the "Clear" operation for AppLockerCSP.

The following fields are available:

- **uri** The URI relative to the %SYSTEM32%\AppLocker folder.

Microsoft.Windows.Security.AppLockerCSP.ClearStart

Start of the "Clear" operation for the AppLockerCSP Node.

Microsoft.Windows.Security.AppLockerCSP.ClearStop

End of the "Clear" operation for the AppLockerCSP node.

The following fields are available:

- **hr** HRESULT reported at the end of the 'Clear' function.

Microsoft.Windows.Security.AppLockerCSP.ConfigManagerNotificationStart

Start of the "ConfigManagerNotification" operation for AppLockerCSP.

The following fields are available:

- **NotifyState** State sent by ConfigManager to AppLockerCSP.

Microsoft.Windows.Security.AppLockerCSP.ConfigManagerNotificationStop

End of the "ConfigManagerNotification" operation for AppLockerCSP.

The following fields are available:

- **hr** HRESULT returned by the ConfigManagerNotification function in AppLockerCSP.

Microsoft.Windows.Security.AppLockerCSP.CreateNodeInstanceParams

Parameters passed to the CreateNodeInstance function of the AppLockerCSP node.

The following fields are available:

- **NodeId** NodeId passed to CreateNodeInstance.
- **nodeOps** NodeOperations parameter passed to CreateNodeInstance.
- **uri** URI passed to CreateNodeInstance, relative to %SYSTEM32%\AppLocker.

Microsoft.Windows.Security.AppLockerCSP.CreateNodeInstanceStart

Start of the "CreateNodeInstance" operation for the AppLockerCSP node.

Microsoft.Windows.Security.AppLockerCSP.CreateNodeInstanceStop

End of the "CreateNodeInstance" operation for the AppLockerCSP node

The following fields are available:

- **hr** HRESULT returned by the CreateNodeInstance function in AppLockerCSP.

Microsoft.Windows.Security.AppLockerCSP.DeleteChildParams

Parameters passed to the DeleteChild function of the AppLockerCSP node.

The following fields are available:

- **child** The child URI of the node to delete.
- **uri** URI relative to %SYSTEM32%\AppLocker.

Microsoft.Windows.Security.AppLockerCSP.DeleteChildStart

Start of the "DeleteChild" operation for the AppLockerCSP node.

Microsoft.Windows.Security.AppLockerCSP.DeleteChildStop

End of the "DeleteChild" operation for the AppLockerCSP node.

The following fields are available:

- **hr** HRESULT returned by the DeleteChild function in AppLockerCSP.

Microsoft.Windows.Security.AppLockerCSP.EnumPolicies

Logged URI relative to %SYSTEM32%\AppLocker, if the Plugin GUID is null, or the CSP doesn't believe the old policy is present.

The following fields are available:

- **uri** URI relative to %SYSTEM32%\AppLocker.

Microsoft.Windows.Security.AppLockerCSP.GetChildNodeNamesParams

Parameters passed to the GetChildNodeNames function of the AppLockerCSP node.

The following fields are available:

- **uri** URI relative to %SYSTEM32%/AppLocker for MDM node.

Microsoft.Windows.Security.AppLockerCSP.GetChildNodeNamesStart

Start of the "GetChildNodeNames" operation for the AppLockerCSP node.

Microsoft.Windows.Security.AppLockerCSP.GetChildNodeNamesStop

End of the "GetChildNodeNames" operation for the AppLockerCSP node.

The following fields are available:

- **child[0]** If function succeeded, the first child's name, else "NA".
- **count** If function succeeded, the number of child node names returned by the function, else 0.
- **hr** HRESULT returned by the GetChildNodeNames function of AppLockerCSP.

Microsoft.Windows.Security.AppLockerCSP.GetLatestId

The result of 'GetLatestId' in AppLockerCSP (the latest time stamped GUID).

The following fields are available:

- **dirId** The latest directory identifier found by GetLatestId.
- **id** The id returned by GetLatestId if id > 0 - otherwise the dirId parameter.

Microsoft.Windows.Security.AppLockerCSP.HResultException

HRESULT thrown by any arbitrary function in AppLockerCSP.

The following fields are available:

- **file** File in the OS code base in which the exception occurs.
- **function** Function in the OS code base in which the exception occurs.
- **hr** HRESULT that is reported.
- **line** Line in the file in the OS code base in which the exception occurs.

Microsoft.Windows.Security.AppLockerCSP.SetValueParams

Parameters passed to the SetValue function of the AppLockerCSP node.

The following fields are available:

- **dataLength** Length of the value to set.
- **uri** The node URI to that should contain the value, relative to %SYSTEM32%\AppLocker.

Microsoft.Windows.Security.AppLockerCSP.SetValueStart

Start of the "SetValue" operation for the AppLockerCSP node.

Microsoft.Windows.Security.AppLockerCSP.SetValueStop

End of the "SetValue" operation for the AppLockerCSP node.

The following fields are available:

- **hr** HRESULT returned by the SetValue function in AppLockerCSP.

Microsoft.Windows.Security.AppLockerCSP.TryRemediateMissingPolicies

EntryPoint of fix step or policy remediation, includes URI relative to %SYSTEM32%\AppLocker that needs to be fixed.

The following fields are available:

- **uri** URI for node relative to %SYSTEM32%/AppLocker.

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_RS1** An ID for the system, calculated by hashing hardware identifiers.
- **DatasourceApplicationFile_RS2** An ID for the system, calculated by hashing hardware identifiers.
- **DatasourceApplicationFile_RS3** The total DecisionApplicationFile objects targeting the next release of Windows on this device.
- **DatasourceApplicationFile_RS4** The count of the number of this particular object type present on this device.
- **DatasourceApplicationFile_RS4Setup** The count of the number of this particular object type present on this device.
- **DatasourceApplicationFile_TH1** The count of the number of this particular object type present on this device.
- **DatasourceApplicationFile_TH2** The count of the number of this particular object type present on this device.
- **DatasourceDevicePnp_RS1** The total DataSourceDevicePnp objects targeting Windows 10 version 1607 on this device.
- **DatasourceDevicePnp_RS2** The count of DatasourceApplicationFile objects present on this machine targeting the next releaseof Windows
- **DatasourceDevicePnp_RS3** The total DatasourceDevicePnp objects targeting the next release of Windows on this device.
- **DatasourceDevicePnp_RS4** The count of the number of this particular object type present on this device.
- **DatasourceDevicePnp_RS4Setup** The count of the number of this particular object type present on this device.
- **DatasourceDevicePnp_TH1** The count of the number of this particular object type present on this device.
- **DatasourceDevicePnp_TH2** The count of the number of this particular object type present on this device.
- **DatasourceDriverPackage_RS1** The total DataSourceDriverPackage objects targeting Windows 10 version 1607 on this device.
- **DatasourceDriverPackage_RS2** The total DataSourceDriverPackage objects targeting Windows 10, version 1703 on this device.
- **DatasourceDriverPackage_RS3** The total DatasourceDriverPackage objects targeting the next release of Windows on this device.
- **DatasourceDriverPackage_RS4** The count of the number of this particular object type present on this device.
- **DatasourceDriverPackage_RS4Setup** The count of the number of this particular object type present on this device.
- **DatasourceDriverPackage_TH1** The count of the number of this particular object type present on this device.
- **DatasourceDriverPackage_TH2** The count of the number of this particular object type present on this device.
- **DataSourceMatchingInfoBlock_RS1** The total DataSourceMatchingInfoBlock objects targeting Windows 10 version 1607 onthis device.
- **DataSourceMatchingInfoBlock_RS2** The count of DatasourceDevicePnp objects present on this machine targeting the nextrele ase of Windows
- **DataSourceMatchingInfoBlock_RS3** The total DataSourceMatchingInfoBlock objects targeting the next release of Windows on this device.
- **DataSourceMatchingInfoBlock_RS4** The count of the number of this particular object type present on this device.
- **DataSourceMatchingInfoBlock_RS4Setup** The count of the number of this particular object type present on this device.
- **DataSourceMatchingInfoBlock_TH1** The count of the number of this particular object type present on this device.
- **DataSourceMatchingInfoBlock_TH2** The count of the number of this particular object type present on this device.
- **DataSourceMatchingInfoPassive_RS1** The total DataSourceMatchingInfoPassive objects targeting Windows 10 version 1607 on this device.
- **DataSourceMatchingInfoPassive_RS2** The count of the number of this particular object type present on this device.
- **DataSourceMatchingInfoPassive_RS3** The total DataSourceMatchingInfoPassive objects targeting the next release of Windows on this device.
- **DataSourceMatchingInfoPassive_RS4** The count of the number of this particular object type present on this device.
- **DataSourceMatchingInfoPassive_RS4Setup** The count of the number of this particular object type present on this device.
- **DataSourceMatchingInfoPassive_TH1** The count of the number of this particular object type present on this device.
- **DataSourceMatchingInfoPassive_TH2** The count of the number of this particular object type present on this device.
- **DataSourceMatchingInfoPostUpgrade_RS1** The total DataSourceMatchingInfoPostUpgrade objects targeting Windows 10versi on 1607 on this device.
- **DataSourceMatchingInfoPostUpgrade_RS2** The count of DatasourceDriverPackage objects present on this machine targetingt he next release of Windows
- **DataSourceMatchingInfoPostUpgrade_RS3** The total DataSourceMatchingInfoPostUpgrade objects targeting the next release of Windows on this device.
- **DataSourceMatchingInfoPostUpgrade_RS4** The count of the number of this particular object type present on this device.
- **DataSourceMatchingInfoPostUpgrade_RS4Setup** The count of the number of this particular object type present on thisdevice.

- **DataSourceMatchingInfoPostUpgrade_TH1** The count of the number of this particular object type present on this device.
- **DataSourceMatchingInfoPostUpgrade_TH2** The count of the number of this particular object type present on this device.
- **DatasourceSystemBios_RS1** The total DatasourceSystemBios objects targeting Windows 10 version 1607 present on thisdevice.
- **DatasourceSystemBios_RS2** The total DatasourceSystemBios objects targeting Windows 10 version 1703 present on thisdevice.
- **DatasourceSystemBios_RS3** The total DatasourceSystemBios objects targeting the next release of Windows on this device.
- **DatasourceSystemBios_RS4** The count of the number of this particular object type present on this device.
- **DatasourceSystemBios_RS4Setup** The count of the number of this particular object type present on this device.
- **DatasourceSystemBios_TH1** The count of the number of this particular object type present on this device.
- **DatasourceSystemBios_TH2** The count of the number of this particular object type present on this device.
- **DecisionApplicationFile_RS1** The count of the number of this particular object type present on this device.
- **DecisionApplicationFile_RS2** The count of the number of this particular object type present on this device.
- **DecisionApplicationFile_RS3** The total DecisionApplicationFile objects targeting the next release of Windows on this device.
- **DecisionApplicationFile_RS4** The count of the number of this particular object type present on this device.
- **DecisionApplicationFile_RS4Setup** The count of the number of this particular object type present on this device.
- **DecisionApplicationFile_TH1** The count of the number of this particular object type present on this device.
- **DecisionApplicationFile_TH2** The count of the number of this particular object type present on this device.
- **DecisionDevicePnp_RS1** The total DecisionDevicePnp objects targeting Windows 10 version 1607 on this device.
- **DecisionDevicePnp_RS2** The count of DataSourceMatchingInfoBlock objects present on this machine targeting the nextrelease of Windows
- **DecisionDevicePnp_RS3** The total DecisionDevicePnp objects targeting the next release of Windows on this device.
- **DecisionDevicePnp_RS4** The count of the number of this particular object type present on this device.
- **DecisionDevicePnp_RS4Setup** The count of the number of this particular object type present on this device.
- **DecisionDevicePnp_TH1** The count of the number of this particular object type present on this device.
- **DecisionDevicePnp_TH2** The count of the number of this particular object type present on this device.
- **DecisionDriverPackage_RS1** The total DecisionDriverPackage objects targeting Windows 10 version 1607 on this device.
- **DecisionDriverPackage_RS2** The count of the number of this particular object type present on this device.
- **DecisionDriverPackage_RS3** The total DecisionDriverPackage objects targeting the next release of Windows on this device.
- **DecisionDriverPackage_RS4** The count of the number of this particular object type present on this device.
- **DecisionDriverPackage_RS4Setup** The count of the number of this particular object type present on this device.
- **DecisionDriverPackage_TH1** The count of the number of this particular object type present on this device.
- **DecisionDriverPackage_TH2** The count of the number of this particular object type present on this device.
- **DecisionMatchingInfoBlock_RS1** The total DecisionMatchingInfoBlock objects targeting Windows 10 version 1607 present ont his device.
- **DecisionMatchingInfoBlock_RS2** The count of DataSourceMatchingInfoPassive objects present on this machine targeting thene xt release of Windows
- **DecisionMatchingInfoBlock_RS3** The total DecisionMatchingInfoBlock objects targeting the next release of Windows on this device.
- **DecisionMatchingInfoBlock_RS4** The count of the number of this particular object type present on this device.
- **DecisionMatchingInfoBlock_RS4Setup** The count of the number of this particular object type present on this device.
- **DecisionMatchingInfoBlock_TH1** The count of the number of this particular object type present on this device.
- **DecisionMatchingInfoBlock_TH2** The count of the number of this particular object type present on this device.
- **DecisionMatchingInfoPassive_RS1** The total DecisionMatchingInfoPassive objects targeting Windows 10 version 1607 on thisd evice.
- **DecisionMatchingInfoPassive_RS2** The count of the number of this particular object type present on this device.
- **DecisionMatchingInfoPassive_RS3** The total DataSourceMatchingInfoPassive objects targeting the next release of Windows on this device.
- **DecisionMatchingInfoPassive_RS4** The count of the number of this particular object type present on this device.
- **DecisionMatchingInfoPassive_RS4Setup** The count of the number of this particular object type present on this device.
- **DecisionMatchingInfoPassive_TH1** The count of the number of this particular object type present on this device.
- **DecisionMatchingInfoPassive_TH2** The count of the number of this particular object type present on this device.
- **DecisionMatchingInfoPostUpgrade_RS1** The total DecisionMatchingInfoPostUpgrade objects targeting Windows 10 version16 07 on this device.
- **DecisionMatchingInfoPostUpgrade_RS2** The count of DataSourceMatchingInfoPostUpgrade objects present on this machineta rgeting the next release of Windows
- **DecisionMatchingInfoPostUpgrade_RS3** The total DecisionMatchingInfoPostUpgrade objects targeting the next release of Windows on this device.
- **DecisionMatchingInfoPostUpgrade_RS4** The count of the number of this particular object type present on this device.

- **DecisionMatchingInfoPostUpgrade_RS4Setup** The count of the number of this particular object type present on this device.
- **DecisionMatchingInfoPostUpgrade_TH1** The count of the number of this particular object type present on this device.
- **DecisionMatchingInfoPostUpgrade_TH2** The count of the number of this particular object type present on this device.
- **DecisionMediaCenter_RS1** The total DecisionMediaCenter objects targeting Windows 10 version 1607 present on this device.
- **DecisionMediaCenter_RS2** The count of DatasourceSystemBios objects present on this machine targeting the next release ofWindows
- **DecisionMediaCenter_RS3** The total DecisionMediaCenter objects targeting the next release of Windows on this device.
- **DecisionMediaCenter_RS4** The count of the number of this particular object type present on this device.
- **DecisionMediaCenter_RS4Setup** The count of the number of this particular object type present on this device.
- **DecisionMediaCenter_TH1** The count of the number of this particular object type present on this device.
- **DecisionMediaCenter_TH2** The count of the number of this particular object type present on this device.
- **DecisionSystemBios_RS1** The total DecisionSystemBios objects targeting Windows 10 version 1607 on this device.
- **DecisionSystemBios_RS2** The total DecisionSystemBios objects targeting Windows 10 version 1703 present on this device.
- **DecisionSystemBios_RS3** The total DecisionSystemBios objects targeting the next release of Windows on this device.
- **DecisionSystemBios_RS4~~InventoryLanguagePack~~** The total DecisionSystemBios~~count of DecisionApplicationFile~~ objects targeting Windows 10 version, 1803 present on this device.
- **DecisionSystemBios_RS4Setup** The total DecisionSystemBios objects~~machine~~ targeting the next release of Windows on thisdevice.
- **DecisionSystemBios_TH1** The count of the number of this particular object type present on this device.
- **DecisionSystemBios_TH2** The count of the number of this particular object type present on this device.
- **InventoryApplicationFile** The count of the number of this particular object type present on this device.
- **InventoryLanguagePack** The count of the number of this particular object type present on this device.
- **InventoryMediaCenter** The count of the number of this particular object type present on this device.
- **InventorySystemBios** The count of the number of this particular object type~~DecisionDevicePnp objects~~ present on this device.
- **InventoryUplevelDriverPackage** The count of~~machine targeting~~ the number~~next release~~ of this particular object type presenton this device.~~Windows~~
- **PCFP** The count of the number of this particular object type~~DecisionDriverPackage objects~~ present on this device.
- **SystemMemory** The count of~~machine targeting~~ the number~~next release~~ of this particular object type present on thisdevice.~~Windows~~
- **SystemProcessorCompareExchange** The count of the number of this particular object type~~DecisionMatchingInfoBlock objects~~present on this device.
- **SystemProcessorLahfSahf** The count of~~machine targeting~~ the number~~next release~~ of this particular object type present on thisdevice.~~Windows~~
- **SystemProcessorNx** The count of the number of this particular object type~~DataSourceMatchingInfoPostUpgrade objects~~present on this device.
- **SystemProcessorPrefetchW** The count of~~machine targeting~~ the number~~next release~~ of this particular object type present onthis device.~~Windows~~
- **SystemProcessorSse2** The count of the number of this particular object type~~DecisionMatchingInfoPostUpgrade objects~~present on this device.
- **SystemTouch** The count of~~machine targeting~~ the number~~next release~~ of this particular object type present on thisdevice.~~Windows~~
- **SystemWim** The count of the number of this particular object type~~DecisionMediaCenter objects~~ present on this device.~~machine targeting the next release of Windows~~
- **SystemWindowsActivationStatus** The count of the number of this particular object type~~DecisionSystemBios objects~~ present on this device.
- **SystemWlan** The count of~~machine targeting~~ the number~~next release~~ of this particular object type present on this device.
- **Wmdrm_RS1** An ID for the system, calculated by hashing hardware identifiers.
- **Wmdrm_RS2~~Windows SystemWlan~~** The count of InventoryLanguagePack~~InventoryApplicationFile~~ objects present on this machine.
- **Wmdrm_RS3** The total Wmdrm objects targeting the next release of Windows on this device.
- **Wmdrm_RS4** The total Wmdrm objects targeting Windows 10, version 1803 present on this device.
- **Wmdrm_RS4Setup** The count of the number of this particular object type present on this device.
- **Wmdrm_TH1** The count of the number of this particular object type present on this device.
- **Wmdrm_TH2** The count of the number of this particular object type present on this device.

Microsoft.Windows.Appraiser.General.DatasourceApplicationFileAdd

Represents the basic metadata about specific application files installed on the system.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **AppraiserVersion** The version of the appraiser file that is generating the events.
- **AvDisplayName** If the app is an anti-virus app, this is its display name.
- **CompatModelIndex** The compatibility prediction for this file.
- **HasCitData** Indicates whether the file is present in CIT data.
- **HasUpgradeExe** Indicates whether the anti-virus app has 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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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 Plug and Play device, to help keep Windows up to date.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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. Microsoft.Windows.Appraiser.General.DatasourceDriverPackageRemove~~ This event~~indicates that the DatasourceDriverPackage object is no longer present. This event~~ includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **AppraiserVersion** The version of the appraiser~~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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

Microsoft.Windows.Appraiser.General.DataSourceMatchingInfoBlockRemove

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

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

Microsoft.Windows.Appraiser.General.DataSourceMatchingInfoPassiveRemove

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

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

Microsoft.Windows.Appraiser.General.DataSourceMatchingInfoPostUpgradeRemove

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

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **AppraiserVersion** The version of the appraiser file that is generating the events.
- **BlockAlreadyInbox** The uplevel runtime block on the file already existed on the current OS.
- **BlockingApplication** Indicates whether there are any application issues that interfere with the 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 that the DecisionApplicationFile object is no longer present.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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?
- **AssociatedDriverWillNotMigrate** Will the driver associated with this plug-and-play device migrate?
- **BlockAssociatedDriver** Should the driver associated with this PNP device be blocked?
- **BlockingDevice** Is this PNP device blocking upgrade?
- **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?
- **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?

Microsoft.Windows.Appraiser.General.DecisionDevicePnpRemove

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

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

Microsoft.Windows.Appraiser.General.DecisionDevicePnpStartSync

The DecisionDevicePnpStartSync~~This~~ event indicates
that a new set of DecisionDevicePnpAdd events will be sent~~the DecisionDevicePnp object is no longer present~~.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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?
- **MediaCenterIndicators** Do any indicators imply that Windows Media Center is in active use?
- **MediaCenterInUse** Is Windows Media Center actively being used?
- **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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

Microsoft.Windows.Appraiser.General.~~EnterpriseScenarioWithDiagTrackServiceRunning This event 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: PCFP An ID for the system calculated by hashing hardware identifiers. Time The client time of the event. 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:

- **NewData** The data in the registry value after the scan completed.

- **OldData** The previous data in the registry value before the scan ran.
- **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.
- **Time** The client time of the event.

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 be part of an antivirus program.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file generating the events.
- **AvDisplayName** If the app is an antivirus app, this is its display name.
- **AvProductState** Indicates whether the antivirus program is turned on and the signatures are up to date.
- **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.
- **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.
- **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.
- **HasUpgradeExe** Indicates whether the antivirus app has an upgrade.exe file.
- **IsAv** Indicates whether the file an antivirus reporting EXE.
- **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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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 that a new set of InventoryApplicationFileAdd events will be sent.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **AppraiserVersion** The version of the Appraiser file that is generating the events.
- **HasLanguagePack** Indicates whether this device has 2 or more language packs.
- **LanguagePackCount** The number of language packs are installed.

Microsoft.Windows.Appraiser.General.InventoryLanguagePackRemove

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

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.
- **BiosDate** The release date of the BIOS in UTC format.
- **biosName** The name field from Win32_BIOS.
- **BiosName** The name field from Win32_BIOS.
- **manufacturer** The manufacturer field from Win32_ComputerSystem.
- **Manufacturer** The manufacturer field from Win32_ComputerSystem.
- **model** The model 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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.
- **Class** The device class from the driver package.
- **ClassGuid** The device class unique ID from the driver package.
- **Date** The date from the driver package.
- **Inbox** Is the driver package of a driver that is included with Windows?
- **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 after it was renamed.
- **Revision** The revision of the driver package.
- **SignatureStatus** Indicates if the driver package is signed. Unknown = 0, Unsigned = 1, Signed = 2.
- **VersionMajor** The major version of the driver package.
- **VersionMinor** The minor version of the driver package.

Microsoft.Windows.Appraiser.General.InventoryUplevelDriverPackageRemove

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

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

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.

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

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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 Windows Imaging Format (WIM) file, to help keep Windows up to date.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

Microsoft.Windows.Appraiser.General.TelemetryRunHealth

This event indicates the parameters and result of a telemetry (diagnostic) 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:

- **AppraiserBranch** The source branch in which the version of Appraiser that is running was built.
- **AppraiserDataVersion** The version of the data files being used by the Appraiser telemetry run.
- **AppraiserProcess** The name of the process that launched Appraiser.
- **AppraiserVersion** The file version (major, minor and build) of the Appraiser DLL, concatenated without dots.
- **AuxFinal** Obsolete, always set to false.
- **AuxInitial** Obsolete, indicates if Appraiser is writing data files to be read by the Get Windows 10 app.
- **DeadlineDate** A timestamp representing the deadline date, which is the time until which appraiser will wait to do a full scan.
- **EnterpriseRun** Indicates if the telemetry run is an enterprise run, which means appraiser was run from the command line with an extra enterprise parameter.
- **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.
- **InboxDataVersion** The original version of the data files before retrieving any newer version.
- **IndicatorsWritten** Indicates if all relevant UEX indicators were successfully written or updated.
- **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.
- **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.
- **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.
- **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.
- **RunDate** The date that the telemetry run was stated, expressed as a filetime.
- **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.
- **RunOnline** Indicates if appraiser was able to connect to Windows Update and theefore is making decisions using up-to-date driver coverage information.
- **RunResult** The hresult of the Appraiser telemetry run.
- **SendingUtc** Indicates if the Appraiser client is sending events during the current telemetry run.
- **StoreHandleIsNotNull** Obsolete, always set to false
- **TelementrySent** Indicates if telemetry was successfully sent.
- **ThrottlingUtc** Indicates if the Appraiser client is throttling its output of CUET events to avoid being disabled. This increases runtime but also telemetry reliability.
- **Time** The client time of the event.
- **VerboseMode** Indicates if appraiser ran in Verbose mode, which is a test-only mode with extra logging.

- **WhyFullSyncWithoutTablePrefix** Indicates the reason or reasons that a full sync was generated.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

Microsoft.Windows.Appraiser.General.WmdrmRemove

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

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

Census events

Census.App

Provides information on IE and Census versions~~This event sends version data about the Apps~~ running on the~~this~~ device~~, to help keep Windows up to date.~~

The following fields are available:

- **AppraiserEnterpriseErrorCode** The error code of the last Appraiser enterprise run.
- **AppraiserErrorCode** The error code of the last Appraiser run.
- **AppraiserRunEndTimeStamp** The end time of the last Appraiser run.

- **AppraiserRunIsInProgressOrCrashed** Flag that indicates if the Appraiser run is in progress or has crashed.
- **AppraiserRunStartTimeStamp** The start time of the last Appraiser run.
- **AppraiserTaskEnabled** Whether the Appraiser task is enabled.
- **AppraiserTaskExitCode** The Appraiser task exist code.
- **AppraiserTaskLastRun** The last runtime for the Appraiser task.
- **CensusVersion** The version of Census that generated the current data for this device.
- **IEVersion** IE~~Retrieves which~~ version~~of Internet Explorer is~~ running on the~~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.
- **InternalBatteryNumberOfCharges** Provides the number of battery charges. This is used when creating new products and validating that existing products meets targeted functionality performance.
- **IsAlwaysOnAlwaysConnectedCapable** Represents whether the battery enables the device to be AlwaysOnAlwaysConnected . Boolean value.

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:

- **AADDeviceId** Azure Active Directory device ID.
- **AzureOSIDPresent** Represents the field used to identify an Azure machine.
- **AzureVMType** Represents whether the instance is Azure VM PAAS, Azure VM IAAS or any other VMs.
- **CDJType** Represents the type of cloud domain joined for the machine.
- **CommercialId** Represents the GUID for the commercial entity which the device is a member of. Will be used to reflect insights back to customers.
- **ContainerType** The type of container, such as process or virtual machine hosted.
- **EnrollmentType** Defines the type of MDM enrollment on the device.
- **HashedDomain** The hashed representation of the user domain used for login.
- **IsCloudDomainJoined** Is this device joined to an Azure Active Directory (AAD) tenant? true/false
- **IsDERequirementMet** Represents if the device can do device encryption.
- **IsDeviceProtected** Represents if Device protected by BitLocker/Device Encryption
- **IsDomainJoined** Indicates whether a machine is joined to a domain.

- **IsEDPEnabled** Represents if Enterprise data protected on the device.
- **IsMDMEnrolled** Whether the device has been MDM Enrolled or not.
- **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~~System Center Configuration Manager (~~ SCCM~~)~~ environment.
- **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.
- **SystemCenterID** The SCCM ID is an anonymized one-way hash of the Active Directory Organization identifier~~.~~

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:

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

- **ActiveMicCount** The number of active microphones attached to the device.
- **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.
- **D3DMaxFeatureLevel** Supported Direct3D version~~DeviceColor Indicates a color of the device~~.
- **DeviceForm** Indicates the form as per the device classification.
- **DeviceName** The device name that is set by the user.
- **DigitizerSupport** Is a digitizer supported?
- **DUID** The device unique ID.
- **Gyroscope** Indicates whether the device has a gyroscope (a mechanical component that measures and maintains orientation).
- **InventoryId** The device ID used for compatibility testing.
- **Magnetometer** Indicates whether the device has a magnetometer (a mechanical component that works like a compass).

- **NFCProximity** Indicates whether the device supports NFC (a set of communication protocols that helps establishcommunication when applicable devices are brought close together.)
- **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.
- **OEMModelBaseBoard** The baseboard model used by the OEM.
- **OEMModelBaseBoardVersion** Differentiates between developer and retail devices.
- **OEMModelName** The device model name.
- **OEMModelNumber** The device model number.
- **OEMModelSKU** The device edition that is defined by the manufacturer.
- **OEMModelSystemFamily** The system family set on the device by an OEM.
- **OEMModelSystemVersion** The system model version set on the device by the OEM.
- **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.
- **PowerPlatformRole** The OEM preferred power management profile. It's used to help to identify the basic form factor of the device.
- **SoCName** The firmware manufacturer of the device.
- **StudyID** Used to identify retail and non-retail device.
- **TelemetryLevel** The telemetry level the user has opted into, such as Basic or Enhanced.
- **TelemetryLevelLimitEnhanced** The telemetry level for Windows Analytics-based solutions.
- **TelemetrySettingAuthority** Determines who set the telemetry level, such as GP, MDM, or the user.
- **TPMManufacturerId** The ID of the TPM manufacturer.
- **TPMManufacturerVersion** The version of the TPM manufacturer.
- **TPMVersion** The supported Trusted Platform Module (TPM) on the device. If no TPM is present, the value is 0.
- **VoiceSupported** Does the device have a cellular radio capable of making voice calls?

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:

- **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.
- **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.
- **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.
- **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.

- **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.
- **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.
- **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.
- **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.
- **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.
- **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.
- **NetworkAdapterGUID** The GUID of the primary network adapter.
- **NetworkCost** Represents the network cost associated with a connection.
- **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.
- **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.

Census.PrivacySettings~~OS~~

This event provides information~~sends data~~ about the device level privacy settings and whether device-level access was granted tothese capabilities. Not all settings are applicable to all devices. Each field records the consent state for the co rresponding privacysetting. The consent state is encoded~~operating system such~~ as a 16-bit signed integer, where the first 8 bits represents the effectiveconsent value~~version~~, and the last 8 bits represent the authority that set the value. The effective consent (first 8 bits) is one of thefollowing values: -3 = unexpected consent value~~locale~~, -2 = value was not requested~~update service configuration~~, -1 = an erroroccurred while attempting to retrieve the value, 0 = undefined, 1 = allow, 2 = deny, 3 = prompt. The consent authority (last 8 bits)is one of the following values: -3 = unexpected authority, -2 = value~~when and how it~~ was not requested~~originally installed~~, -1 = anerror occurred while attempting to retrieve the value, 0 = system, 1 = a higher authority (a gating setting, the system-wide setting,or a group policy), 2 = advertising ID group policy, 3 = advertising ID policy for child account, 4 = privacy setting provider doesn'tknow the actual consent authority, 5 = consent was not configured and~~whether it is~~ a default set in code was used~~virtual device~~, 6= system default, 7 = organization policy, 8 = OneSettings~~to help keep Windows up to date~~.

The following fields are available:

- **Activity** Current state of~~ActivationChannel Retrieves~~ the activity history setting~~retail license key or Volume license key for a mach ine~~.
- **ActivityHistoryCloudSync** Current state of~~CompactOS Indicates if~~ the activity history cloud sync setting~~Compact OS feature fro m Win10 is enabled~~.
- **ActivityHistoryCollection** Current state of~~DeveloperUnlockStatus Represents if a device has been developer unlocked by~~ the ac tivity history collection setting~~user or Group Policy~~.
- **AdvertisingId** Current state of~~DeviceTimeZone The time zone that is set on~~ the advertising ID setting~~device~~.
- **AppDiagnostics** Current state of~~Example: Pacific Standard Time GenuineState Retrieves~~ the app diagnostics setting~~ID Value spec ifying the OS Genuine check~~.
- **Appointments** Current state of~~InstallationType Retrieves~~ the calendar setting.
- **Bluetooth** Current state~~type~~ of the Bluetooth capability setting~~OS installation~~.
- **BluetoothSync** Current state of~~(Clean, Upgrade, Reset, Refresh, Update). InstallLanguage The first language installed on~~ the Blue tooth sync capability setting~~user machine~~.
- **BroadFileSystemAccess** Current state of~~IsDeviceRetailDemo Retrieves if~~ the broad file system access setting~~device is running in demo mode~~.
- **CellularData** Current state of~~IsEduData Returns Boolean if~~ the cellular~~education~~ data capability setting~~policy is enabled~~.
- **Chat** Current state of the chat setting~~IsPortableOperatingSystem Retrieves whether OS is running Windows To Go IsSecureBootEnabled Retrieves whether Boot chain is signed under UEFI~~.

- **Contacts** Current stateLanguagePacks The list oflanguage packages installed on the contacts settingdevice.
- **DocumentsLibrary** Current state ofLicenseStateReason 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 bythe documents library settingMS store.
- **Email** Current state ofOA3xOriginalProductKey Retrieves the email settingLicense key stamped by the OEM to the machine.
- **FindMyDevice** Current state ofOSEdition Retrieves the "find my device" setting.
- **GazeInput** Current stateversion of the gaze input settingcurrent OS.
- **HumanInterfaceDevice** Current stateOSInstallType Retrieves a numeric description ofwhat install was used on the humaninterface device settingi.
- **InkTypeImprovement** Current state of the improve inking and typing settinge.
- **Location** Current stateclean, upgrade, refresh, reset, etc OSOOBEDateTime Retrieves Out ofBox Experience (OOBE) Date in Coordinated Universal Time (UTC). OSSKU Retrieves the location setting.
- **LocationHistory** Current stateFriendly Name of the location history settingOS Edition.
- **LocationHistoryCloudSync** Current state ofOSSubscriptionStatus Represents the location history cloud sync settingexisting status for enterprise subscription feature for PRO machines.
- **LocationHistoryOnTimeline** Current state ofOSSubscriptionTypeId Returns boolean for enterprise subscription feature for selected PRO machines. OSTimeZoneBiasInMins Retrieves the location historytime zone set on timeline settingmachine.
- **Microphone** Current state ofOSUILocale Retrieves the microphone setting.
- **PhoneCall** Current statelocale of the phone call setting.
- **PhoneCallHistory** Current state ofUI that is currently used by the call history settingOS.
- **PicturesLibrary** Current state ofProductActivationResult Returns Boolean if the pictures library settingOS Activation was successful.
- **Radios** Current state ofProductActivationTime Returns the radios settingOS Activation time for tracking piracy issues.
- **SensorsCustom** Current state ofProductKeyID2 Retrieves the custom sensor setting.
- **SerialCommunication** Current state ofLicense key if the serial communication settingmachine is updated with a new license key.
- **Sms** Current state ofRACw7Id Retrieves the text messaging settingMicrosoft Reliability Analysis Component (RAC) Win7 Identifier.
- **SpeechPersonalization** Current state of the speech services settingRAC is used to monitor and analyze system usage and reliability.
- **USB** Current state ofServiceMachineIP Retrieves the USB setting.
- **UserAccountInformation** Current stateIP address of the account information settingKMS host used for anti-piracy.
- **UserDataTasks** Current state ofServiceMachinePort Retrieves the tasks setting.
- **UserNotificationListener** Current stateport of the notifications settingKMS host used for anti-piracy.
- **VideosLibrary** Current state ofServiceProductKeyID Retrieves the videos library setting.
- **Webcam** Current stateLicense key of the camera settingKMS SharedPCMode Returns Boolean for education devices used as shared cart Signature Retrieves if it is a signature machine sold by Microsoft store.
- **WiFiDirect** Current state ofSLICStatus Whether a SLIC table exists on the Wi-Fi direct settingdevice.SLICVersion Returns OS type/version from SLIC table.

Census.Processor

Provides information on several importantThis event sends data points about Processor settingsthe processor (architecture, speed, number of cores, manufacturer, and model number), to help keep Windows up to date.

The following fields are available:

- **KvaShadow** Microcode info of the processor.
- **MMSettingOverride** Microcode setting of the processor.
- **MMSettingOverrideMask** Microcode setting override of the processor.
- **PreviousUpdateRevision** Previous microcode revision
- **ProcessorArchitecture** Retrieves the processor architecture of the installed operating system.
- **ProcessorClockSpeed** ClockRetrieves the clock speed of the processor in MHz.
- **ProcessorCores** NumberRetrieves the number of logical cores in the processor.
- **ProcessorIdentifier** Processor IdentifierThe processor identifier of a manufacturer.
- **ProcessorManufacturer** NameRetrieves the name of the processorprocessor's manufacturer.
- **ProcessorModel** NameRetrieves the name of the processor model.

- **ProcessorPhysicalCores** Number of physical cores in the processor.
- **ProcessorUpdateRevision** Microcode revision
- **ProcessorUpdateStatus** Enum value that represents the processor microcode load status
- **SocketCount** CountNumber ofphysical CPU sockets.
- **SpeculationControl** If the system has enabled protections needed to validate the speculation control vulnerability.

Census.Security

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

The following fields are available:

- **AvailableSecurityProperties** This field helps to enumerate and report state on the relevant security properties for DeviceGuard.
- **CGRunning** Credential Guard isolates and hardens key system and user secrets against compromise, helping to minimize theimp act and breadth of a Pass the Hash style attack in the event that malicious code is already running via a local or networkbased ve ctormachine. This field tells if Credential Guard is running.
- **DGState** This field summarizes the Device Guard state.
- **HVCIRunning** Hypervisor Code Integrity (HVCI) enables Device Guard to help protect kernel mode processes and drivers fromvu lnerability exploits and zero days. HVCI uses the processor's functionality to force all software running in kernel mode tosafely all ocate memory. This field tells if HVCI is running.
- **IsSawGuest** Indicates whether the device is running as a Secure Admin Workstation Guest.
- **IsSawHost** Indicates whether the device is running as a Secure Admin Workstation Host.
- **RequiredSecurityProperties** Describes the required security properties to enable virtualization-based security.
- **SecureBootCapable** Systems that support Secure Boot can have the feature turned off via BIOS. This field tells if the system isca pable of running Secure Boot, regardless of the BIOS setting.
- **SModeState** The Windows S mode trail state.
- **VBSState** Virtualization-based security (VBS) uses the hypervisor to help protect the kernel and other parts of the operatingsystem. Credential Guard and Hypervisor Code Integrity (HVCI) both depend on VBS to isolate/protect secrets, and kernel-mode code integrity validation. VBS has a tri-state that can be Disabled, Enabled, or Running.

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.
- **KeyVer** Version information for the census speech event.
- **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 administratoradmininistrator (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.
- **SpeechServicesValueSource** Indicates the deciding factor for the effective online speech recognition privacy policy settings:rem ote admin, local admin, or user preference.

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.
- **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).
- **SystemVolumeTotalCapacity** Retrieves the size of the partition that the System volume is installed on in MB.

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:

- **DefaultApp** The current uer's default program selected for the following extension or protocol: .html, .htm, .jpg, .jpeg, .png, .mp3, .mp4, .mov, .pdf.
- **DefaultBrowserProgId** The ProgramId of the current user's default browser.

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
- **NumberofExternalDisplays** Retrieves the number of external displays connected to the machine
- **NumberofInternalDisplays** Retrieves the number of internal displays in a 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.
- **DisplayLanguage** The current user preferred Windows Display Language.
- **HomeLocation** The current user location, which is populated using GetUserGeoId() function.
- **KeyboardInputLanguages** The Keyboard input languages installed on the device.
- **SpeechInputLanguages** The Speech Input languages installed on the device.

Census.UserPrivacySettings

This event provides information about the current users privacy settings and whether device-level access was granted to thesecapabilities. Not all settings are applicable to all devices. Each field records the consent state for the corresponding privacy setting.The consent state is encoded as a 16-bit signed integer, where the first 8 bits represents the effective consent value, and the last 8bits represents the authority that set the value. The effective consent is one of the following values: -3 = unexpected consent value,-2 = value was not requested, -1 = an error occurred while attempting to retrieve the value, 0 = undefined, 1 = allow, 2 = deny, 3 =prompt. The consent authority is one of the following values: -3 = unexpected authority, -2 = value was not requested, -1 = anerror occurred while attempting to retrieve the value, 0 = user, 1 = a higher authority (a gating setting, the system-wide setting, ora group policy), 2 = advertising ID group policy, 3 = advertising ID policy for child account, 4 = privacy setting provider doesn'tknow the actual consent authority, 5 = consent was not configured and a default set in code was used, 6 = system default, 7 =organization policy, 8 = OneSettings.

The following fields are available:

- **Activity** Current state of the activity history setting.
- **ActivityHistoryCloudSync** Current state of the activity history cloud sync setting.
- **ActivityHistoryCollection** Current state of the activity history collection setting.
- **AdvertisingId** Current state of the advertising ID setting.
- **AppDiagnostics** Current state of the app diagnostics setting.
- **Appointments** Current state of the calendar setting.
- **Bluetooth** Current state of the Bluetooth capability setting.
- **BluetoothSync** Current state of the Bluetooth sync capability setting.
- **BroadFileSystemAccess** Current state of the broad file system access setting.
- **CellularData** Current state of the cellular data capability setting.
- **Chat** Current state of the chat setting.
- **Contacts** Current state of the contacts setting.
- **DocumentsLibrary** Current state of the documents library setting.
- **Email** Current state of the email setting.
- **GazeInput** Current state of the gaze input setting.
- **HumanInterfaceDevice** Current state of the human interface device setting.
- **InkTypeImprovement** Current state of the improve inking and typing setting.
- **InkTypePersonalization** Current state of the inking and typing personalization setting.
- **Location** Current state of the location setting.
- **LocationHistory** Current state of the location history setting.
- **LocationHistoryCloudSync** Current state of the location history cloud synchronization setting.
- **LocationHistoryOnTimeline** Current state of the location history on timeline setting.
- **Microphone** Current state of the microphone setting.
- **PhoneCall** Current state of the phone call setting.
- **PhoneCallHistory** Current state of the call history setting.
- **PicturesLibrary** Current state of the pictures library setting.
- **Radios** Current state of the radios setting.
- **SensorsCustom** Current state of the custom sensor setting.
- **SerialCommunication** Current state of the serial communication setting.
- **Sms** Current state of the text messaging setting.
- **SpeechPersonalization** Current state of the speech services setting.
- **USB** Current state of the USB setting.
- **UserAccountInformation** Current state of the account information setting.
- **UserDataTasks** Current state of the tasks setting.
- **UserNotificationListener** Current state of the notifications setting.
- **VideosLibrary** Current state of the videos library setting.
- **Webcam** Current state of the camera setting.
- **WiFiDirect** Current state of the Wi-Fi direct setting.

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:

- **CloudService** Indicates which cloud service, if any, that this virtual machine is running within.
- **HyperVisor** Retrieves whether the current OS is running on top of a Hypervisor.
- **IOMMUPresent** Represents if an input/output memory management unit (IOMMU) is present.
- **IsVDI** Is the device using Virtual Desktop Infrastructure?
- **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.
- **SLATSupported** Represents whether Second Level Address Translation (SLAT) is supported by the hardware.
- **VirtualizationFirmwareEnabled** Represents whether virtualization is enabled in the firmware.

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:

- **AppraiserGatedStatus** Indicates whether a device has been gated for upgrading.
- **AppStoreAutoUpdate** Retrieves the Appstore settings for auto upgrade. (Enable/Disabled).
- **AppStoreAutoUpdateMDM** Retrieves the App Auto Update value for MDM: 0 - Disallowed. 1 - Allowed. 2 - Not configured. Default: [2] Not configured
- **AppStoreAutoUpdatePolicy** Retrieves the Microsoft Store App Auto Update group policy setting
- **DelayUpgrade** Retrieves the Windows upgrade flag for delaying upgrades.
- **OSAssessmentFeatureOutOfDate** How many days has it been since a the last feature update was released but the device didno t 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 didnot install it?
- **OSAssessmentReleaseInfoTime** The freshness of release information used to perform an assessment.
- **OSRollbackCount** The number of times feature updates have rolled back on the device.
- **OSRolledBack** A flag that represents when a feature update has rolled back during setup.
- **OSUninstalled** A flag that represents when a feature update is uninstalled on a device .
- **OSWUAutoUpdateOptions** Retrieves the auto update settings on the device.
- **OSWUAutoUpdateOptionsSource** The source of auto update setting that appears in the OSWUAutoUpdateOptions field. Forex ample: Group Policy (GP), Mobile Device Management (MDM), and Default.
- **UninstallActive** A flag that represents when a device has uninstalled a previous upgrade recently.
- **UpdateServiceURLConfigured** Retrieves if the device is managed by Windows Server Update Services (WSUS).
- **WUDeferUpdatePeriod** Retrieves if deferral is set for Updates.
- **WUDeferUpgradePeriod** Retrieves if deferral is set for Upgrades.
- **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.
- **WUMachineId** Retrieves the Windows Update (WU) Machine Identifier.
- **WUPauseState** Retrieves WU setting to determine if updates are paused.
- **WUServer** Retrieves the HTTP(S) URL of the WSUS server that is used by Automatic Updates and API callers (by default).

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:

- **XboxConsolePreferredLanguage** Retrieves the preferred language selected by the user on Xbox console.
- **XboxConsoleSerialNumber** Retrieves the serial number of the Xbox console.

- **XboxLiveDeviceId** Retrieves the unique device ID of the console.
- **XboxLiveSandboxId** Retrieves the developer sandbox ID if the device is internal to Microsoft.

Common data extensions

Common Data Extensions.app

Describes the properties of the running application. This extension could be populated by a client app or a web app.

The following fields are available:

- **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.
- **env** The environment from which the event was logged.
- **expId** Associates a flight, such as an OS flight, or an experiment, such as a web site UX experiment, with an event.
- **id** 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.
- **locale** The locale of the app.
- **name** The name of the app.
- **userId** The userID as known by the application.
- **ver** Represents the version number of the application. Used to understand errors by Version, Usage by Version across an app.

Common Data Extensions.container

Describes the properties of the container for events logged within a container.

The following fields are available:

- **epoch** An ID that's incremented for each SDK initialization.
- **localId** The device ID as known by the client.
- **osVer** The operating system version.
- **seq** An ID that's incremented for each event.
- **type** The container type. Examples: Process or VMHost

Common Data Extensions.cs

Describes properties related to the schema of the event.

The following fields are available:

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

Common Data Extensions.device

Describes the device-related fields.

The following fields are available:

- **deviceClass** The~~Represents the classification of the~~ device classification.~~, the device family".~~ For example, Desktop, Server, or Mobile.
- **localId** A locally-defined~~Represents a locally defined~~ unique ID for the device. This is~~,~~ not the human-readable~~human-readable~~device name. Most likely equal to the value stored at HKLM\Software\Microsoft\SQMClient\MachineId
- **make** Device manufacturer.
- **model** Device model.

Common Data Extensions.Envelope

Represents an envelope that contains all of the common data extensions.

The following fields are available:

~~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 and 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.
- **data** Represents the optional unique diagnostic data for a particular event schema.~~epoch ID used to help distinguish events in the sequence by indicating the current boot session.~~
- **ext_app** Describes the properties of the running application. This extension could be populated by either a client app or a web app. See Common Data Extensions.app.
- **ext_container** Describes the properties of the container for events logged within a container. See Common Data Extensions.container.
- **ext_cs** Describes properties related to the schema of the event. See Common Data Extensions.cs.
- **ext_device** Describes the device-related fields. See Common Data Extensions.device.
- **ext_os** Describes the operating system properties that would be populated by the client. See Common Data Extensions.os.
- **ext_receipts** Describes the fields related to time as provided by the client for debugging purposes. See Common DataExtensions .receipts.
- **ext_sdk** Describes the fields related to a platform library required for a specific SDK. See Common Data Extensions.sdk.
- **ext_user** Describes the fields related to a user. See Common Data Extensions.user.
- **ext_utc** Describes the fields that might be populated by a logging library on Windows. See Common Data Extensions.utc.
- **ext_xbl** Describes the fields related to XBOX Live. See Common Data Extensions.xbl.
- **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.
- **iKey** Represents an ID for applications or other logical groupings of events.
- **name** Represents the uniquely qualified name for the event.~~os The operating system name. osVer The operating system version.~~
- **popSample** Represents the effective sample rate for this event at the time it was generated by a client.~~seqNum Used to track the absolute order of uploaded events. tags A header for semi-managed extensions.~~
- **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.
- **ver** Represents the major and minor version of the extension.

Common Data Extensions.os

Describes some properties of the operating system.

The following fields are available:

- **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.
- **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.
- **name** Represents the operating system name.
- **ver** Represents the major and minor version of the extension.

Common Data Extensions.receipts

Represents various time information as provided by the client and helps for debugging purposes.

The following fields are available:

- **originalTime** The original event time.
- **uploadTime** The time the event was uploaded.

Common Data Extensions.sdk

Used by platform specific libraries to record fields that are required for a specific SDK.

The following fields are available:

- **epoch** An ID that is incremented for each SDK initialization.
- **installId** An ID that's created during the initialization of the SDK for the first time.
- **libVer** The SDK version.
- **seq** An ID that is incremented for each event.

Common Data Extensions.user

Describes the fields related to a user.

The following fields are available:

- **authId** This is an ID of the user associated with this event that is deduced from a token such as a Microsoft Account ticket or an XBOX token.
- **locale** The language and region.
- **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.utc

Describes the properties that could be populated by a logging library on Windows.

The following fields are available:

- **aId** Represents the ETW ActivityId. Logged via TraceLogging or directly via ETW.
- **bSeq** Upload buffer sequence number in the format: buffer identifier:sequence number
- **cat** Represents a bitmask of the ETW Keywords associated with the event.
- **cpId** The composer ID, such as Reference, Desktop, Phone, Holographic, Hub, IoT Composer.
- **epoch** Represents the epoch and seqNum fields, which help track how many events were fired and how many events wereuploaded, and enables identification of data lost during upload and de-duplication of events on the ingress server.
- **flags** Represents the bitmap that captures various Windows specific flags.
- **mon** Combined monitor and event sequence numbers in the format: monitor sequence : event sequence
- **op** Represents the ETW Op Code.
- **raId** Represents the ETW Related ActivityId. Logged via TraceLogging or directly via ETW.
- **seq** Represents the sequence field used to track absolute order of uploaded events. It is an incrementing identifier for eachevent added to the upload queue.~~sqmId~~ The Sequence helps track how many events were fired and how many events wereuploaded and enables identification of data lost during upload and de-duplication of events on the ingress server~~Windows SQM ID~~.
- **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.~~tickets An array of strings that refer back to a key in the X-Tickets http header that the client uploaded along with a batch of events.~~

Common Data Extensions.xbl

Describes the fields that are related to XBOX Live.

The following fields are available:

- **claims** Any additional claims whose short claim name hasn't been added to this structure.
- **did** XBOX device ID
- **dty** XBOX device type
- **dvr** The version of the operating system on the device.
- **eid** A unique ID that represents the developer entity.
- **exp** Expiration time
- **ip** The IP address of the client device.
- **nbf** Not before time
- **pid** A comma separated list of PUIDs listed as base10 numbers.
- **sbx** XBOX sandbox identifier
- **sid** The service instance ID.
- **sty** The service type.
- **tid** The XBOX Live title ID.
- **tvr** The XBOX Live title version.
- **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.
- **xid** A list of base10-encoded XBOX User IDs.

Common data fields

Ms.Device.DeviceInventoryChange

Describes the installation state for all hardware and software components available on a particular device.

The following fields are available:

- **action** The change that was invoked on a device inventory object.
- **inventoryId** Device ID used for Compatibility testing
- **objectInstanceId** Object identity which is unique within the device scope.
- **objectType** Indicates the object type that the event applies to.
- **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.

Component-based servicing~~Diagnostic data~~ events

CbsServicingProvider~~TelClientSynthetic~~.CbsLateAcquisition~~AuthorizationInfo_RuntimeTransition~~

This event sends
data to indicate if some Operating System packages could not be updated as part~~indicating that a device has undergone a change~~ of an upgrade~~telemetry opt-in level detected at UTC startup~~, to help keep Windows up to date.

The following fields~~telemetry opt-in level signals what data we~~ are available:

- **Features** The list of feature packages that could not be updated.
- **RetryID** The ID identifying the retry attempt~~allowed~~ to update the listed packages~~collect~~.

Deployment extensions

DeploymentTelemetry.Deployment_End

This event indicates that a Deployment 360 API has completed.

The following fields are available:

- **ClientId** Client ID of the~~CanAddMsaToMsTelemetry True if UTC is allowed to add MSA~~ user utilizing~~identity onto telemetry from~~ the D360 API~~OS provider groups~~.
- **ErrorCode** Error code of action~~CanCollectAnyTelemetry True if UTC is allowed to collect non-OS telemetry~~.

- **FlightId** The specific ID of the Windows Insider build the device~~Non-OS telemetry~~ is getting~~responsible for providing its own opt-in mechanism~~.
- **Mode** Phase in upgrade~~CanCollectCoreTelemetry True if UTC is allowed to collect data which is tagged with both MICROSOFT_KEYWORD_CRITICAL_DATA and MICROSOFT_EVENTTAG_CORE_DATA~~.
- **RelatedCV** The correction vector~~CanCollectHeartbeats True if UTC is allowed to collect heartbeats. CanCollectOsTelemetry True if UTC is allowed to collect telemetry from the OS provider groups~~ (CV) of any other related events
- **Result** End result of the action~~often called Microsoft Telemetry). CanPerformDiagnosticEscalations True if UTC is allowed to perform all scenario escalations~~.

DeploymentTelemetry~~CanPerformScripting True if UTC is allowed to perform scripting~~.Deployment_SetupBoxLaunch

This event indicates that the Deployment 360 APIs have launched Setup Box~~CanPerformTraceEscalations True if UTC is allowed to perform scenario escalations with tracing actions~~.

The following fields are available:

- **ClientId** The client ID of~~CanReportScenarios True if UTC is allowed to load and report scenario completion, failure, and cancellation events. PreviousPermissions Bitmask representing~~ the user utilizing~~previously configured permissions since~~ the D360 API~~telemetry opt-in level was last changed~~.
- **FlightId** The specific ID of the Windows Insider build the device~~TransitionFromEverythingOff True if this transition~~ is getting~~moving from not allowing core telemetry to allowing core telemetry~~.
- **Quiet** Whether Setup will run in quiet mode or full mode.
- **RelatedCV** The correlation vector (CV) of any other related events.
- **SetupMode** The current setup phase.

DeploymentTelemetry~~TelClientSynthetic~~.Deployment~~AuthorizationInfo~~_SetupBoxResult~~Startup~~

This
event indicates~~sends data indicating~~ that the Deployment 360 APIs have received a return from Setup Box~~device has undergone a change of telemetry opt-in level detected at UTC startup, to help keep Windows up to date.The telemetry opt-in level signals what data we are allowed to collect.~~

The following fields are available:

- **ClientId** Client ID of the~~CanAddMsaToMsTelemetry True if UTC is allowed to add MSA~~ user utilizing~~identity onto telemetry from~~ the D360 API~~OS provider groups~~.
- **ErrorCode** Error code of the action~~CanCollectAnyTelemetry True if UTC is allowed to collect non-OS telemetry~~.
- **FlightId** The specific ID of the Windows Insider build the device~~Non-OS telemetry~~ is getting~~responsible for providing its own opt-in mechanism~~.
- **Quiet** Indicates whether Setup will run in quiet mode or full mode~~CanCollectCoreTelemetry True if UTC is allowed to collect data which is tagged with both MICROSOFT_KEYWORD_CRITICAL_DATA and MICROSOFT_EVENTTAG_CORE_DATA~~.
- **RelatedCV** The correlation vector~~CanCollectHeartbeats True if UTC is allowed to collect heartbeats. CanCollectOsTelemetry True if UTC is allowed to collect telemetry from the OS provider groups~~ (CV) of any other related events~~often called Microsoft Telemetry). CanPerformDiagnosticEscalations True if UTC is allowed to perform all scenario escalations~~.
- **SetupMode** The current Setup phase~~CanPerformScripting True if UTC is allowed to perform scripting~~.

DeploymentTelemetry~~CanPerformTraceEscalations True if UTC is allowed to perform scenario escalations with tracing actions.~~Deployment_Start

This event indicates that a Deployment 360 API has been called~~CanReportScenarios True if UTC is allowed to load and report scenario completion, failure, and cancellation events~~.

The following fields are available:

- **ClientId** Client ID ofPreviousPermissions Bitmask representing the user utilizingpreviously configured permissions since the D360 APItelemetry client was last started.
- **FlightId** The specific ID of the Windows Insider build the deviceTransitionFromEverythingOff True if this transition is gettingmoving from not allowing core telemetry to allowing core telemetry.
- **Mode** The current phase of the upgrade.
- **RelatedCV** The correlation vector (CV) of any other related events.

Diagnostic data events

TelClientSynthetic.AbnormalShutdownConnectivityHeartBeat_0

This event sends data
about boot IDs for whichthe 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 normal clean shutdownConnectivity Heartbeat event wasfired in the past 24 hours, and if not observed,it fires an event. A Connectivity Heartbeat event also fires when a device recovers from costed network to help keepWindows up to datefree network.

The following fields are available:

- **AbnormalShutdownBootId** BootId of the abnormal shutdown being reported by this event.
- **AcDcStateAtLastShutdown** Identifies if the device was on battery or plugged in.
- **BatteryLevelAtLastShutdown** TheCensusExitCode Returns last recorded battery levelexecution codes from census client run.
- **BatteryPercentageAtLastShutdown** The battery percentage at the last shutdown.
- **CrashDumpEnabled** Are crash dumps enabled?
- **CumulativeCrashCount** Cumulative count of operating system crashes since the BootId reset.
- **CurrentBootId** BootId at the time the abnormal shutdown event was being reported.
- **Firmwaredata->ResetReasonEmbeddedController** The reset reason that was supplied by the firmware.
- **Firmwaredata->ResetReasonEmbeddedControllerAdditional** Additional data relatedCensusStartTime Returns timestamp corresponding to reset reason provided by the firmware.
- **Firmwaredata->ResetReasonPch** The reset reason that was supplied by the hardware.
- **Firmwaredata->ResetReasonPchAdditional** Additional data related to the reset reason supplied by the hardware.
- **Firmwaredata->ResetReasonSupplied** Indicates whether the firmware supplied any reset reason or not.
- **FirmwareType** ID of the FirmwareType as enumerated in DimFirmwareType.
- **HardwareWatchdogTimerGeneratedLastReset** Indicates whether the hardware watchdog timer caused the last resetsuccessful census run.
- **HardwareWatchdogTimerPresent** Indicates whether hardware watchdog timer was present or not.
- **LastBugCheckBootId** bootId ofCensusTaskEnabled Returns Boolean value for the last captured crash.
- **LastBugCheckCode** Code that indicates the type of error.
- **LastBugCheckContextFlags** Additional crash dump settings.
- **LastBugCheckOriginalDumpType** The type of crash dump the system intended to save.
- **LastBugCheckOtherSettings** Other crash dump settings.
- **LastBugCheckParameter1** The first parameter with additional infocensus task (Enable/Disable) on the type of the errorclient machine.
- **LastBugCheckProgress** Progress towards writing outLastConnectivityLossTime Retrieves the last crash dump.
- **LastBugCheckVersion** The version of the information struct written during the crash.
- **LastSuccessfullyShutdownBootId** BootId of the last fully successful shutdown.
- **LongPowerButtonPressDetected** Identifies if the user was pressing and holding power button.
- **OOBEInProgress** Identifies if OOBE is running.
- **OSSetupInProgress** Identifies if the operating system setup is running.
- **PowerButtonCumulativePressCount** How many times has the power button been pressed?
- **PowerButtonCumulativeReleaseCount** How many times has the power button been released?
- **PowerButtonErrorCount** Indicates the number of times there was an error attempting to record power button metrics.
- **PowerButtonLastPressBootId** BootId of the last time the power button was presseddevice lost free network.
- **PowerButtonLastPressTime** Date and time ofLastConntectivityLossTime Retrieves the last time the power button was pressed.
- **PowerButtonLastReleaseBootId** BootId of the last time the power button was released.

- **PowerButtonLastReleaseTime** Date and time of the last time the power button was released.
- **PowerButtonPressCurrentCsPhase** Represents the phase of Connected Standby exit when the power button was pressed.
- **PowerButtonPressIsShutdownInProgress** Indicates whether a system shutdown was in progress at the last time the powerbutt on was pressed.
- **PowerButtonPressLastPowerWatchdogStage** Progress while the monitor is being turned on.
- **PowerButtonPressPowerWatchdogArmed** Indicates whether or not the watchdog for the monitor was active at the time of the last power button press.
- **ShutdownDeviceType** Identifies who triggered a shutdown. Is it because of battery, thermal zones, or through a Kernel API.
- **SleepCheckpoint** Provides the last checkpoint when there is a failure during a sleep transition.
- **SleepCheckpointSource** Indicates whether the source is the EFI variable or bootstat file.
- **SleepCheckpointStatus** Indicates whether the checkpoint information is valid.
- **StaleBootStatData** Identifies if the data from bootstat is stale.
- **TransitionInfoBootId** BootId of the captured transition info.
- **TransitionInfoCSCount** I number of times the system transitioned from Connected Standby mode.
- **TransitionInfoCSEntryReason** Indicates the reason the device last entered Connected Standby mode~~lost free network~~.
- **TransitionInfoCSExitReason** Indicates~~NetworkState Retrieves~~ the reason the device last exited Connected Standbymode~~networ k state: 0 = No network~~.
- **TransitionInfoCSInProgress** At the time the last marker was saved, the system was in or entering Connected Standby mode~~1 = Restricted network~~.
- **TransitionInfoLastReferenceTimeChecksum** The checksum of TransitionInfoLastReferenceTimestamp,
- **TransitionInfoLastReferenceTimestamp** The date and time that the marker was last saved~~2 = Free network~~.
- **TransitionInfoLidState** Describes~~NoNetworkTime Retrieves~~ the state of the laptop lid.
- **TransitionInfoPowerButtonTimestamp** The date and time of~~spent with no network (since~~ the last time the power button waspressed.
- **TransitionInfoSleepInProgress** At the time the last marker was saved, the system was~~)~~ in or entering sleep mode~~seconds~~.
- **TransitionInfoSleepTranstionsToOn** Total number of times~~RestrictedNetworkTime Retrieves~~ the device transitioned from sleep mode.
- **TransitionInfoSystemRunning** At the time the last marker was saved, the device was running.
- **TransitionInfoSystemShutdownInProgress** Indicates whether a device shutdown was in progress when the power button wasp ressed.
- **TransitionInfoUserShutdownInProgress** Indicates whether a user shutdown was in progress when the power button waspresse d.
- **TransitionLatestCheckpointId** Represents a unique identifier for a checkpoint during the device state transition.
- **TransitionLatestCheckpointSeqNumber** Represents the chronological sequence number of the checkpoint.
- **TransitionLatestCheckpointType** Represents the type of the checkpoint, which can be the start of a phase, end of a phase, orju st informational.
- **VirtualMachineId** If the operating system is~~spent~~ on a virtual Machine, it gives the virtual Machine ID~~metered~~ (GUID~~cost restricted~~) that can be used to correlate events on the host~~ne twork in seconds~~.

TelClientSynthetic.HeartBeat_5

This event sends data about the health and quality of the diagnostic 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:

- **AgentConnectionErrorsCount** Number~~The number~~ of non-timeout errors associated with the host/agent channel.
- **CensusExitCode** The last exit code of the Census task.
- **CensusStartTime** Time~~The time~~ of~~the~~ last Census run.
- **CensusTaskEnabled** True if~~Indicates whether~~ Census is enabled, false otherwise.
- **CompressedBytesUploaded** Number of compressed bytes uploaded.
- **ConsumerDroppedCount** Number~~The number~~ of events dropped at~~by the~~ consumer layer of~~the~~ telemetry client.
- **CriticalDataDbDroppedCount** Number~~The number~~ of critical data sampled events~~that were~~ dropped at the database layer.
- **CriticalDataThrottleDroppedCount** The number of critical data sampled events that were dropped because of throttling.
- **CriticalOverflowEntersCounter** Number~~The number~~ of times~~a~~ critical overflow mode was entered in~~into the~~ event DB~~database~~.
- **DbCriticalDroppedCount** Total~~The total~~ number of dropped critical events in~~the~~ event DB~~database~~.

- **DbDroppedCount** NumberThe number of eventsthat were dropped due to DB fullnessbecause the database was full.
- **DbDroppedFailureCount** NumberDecodingDroppedCount The number of events dropped due to DB failures.
- **DbDroppedFullCount** Numberbecause of events dropped due to DB fullness.
- **DecodingDroppedCount** Number of events dropped due to decoding failures.
- **EnteringCriticalOverflowDroppedCounter** NumberThe number of eventsthat was dropped due tobecause a critical overflow mode beingwas initiated.
- **EtwDroppedBufferCount** NumberThe number of buffers dropped in the UTCCUET ETW session.
- **EtwDroppedCount** NumberThe number of events dropped atby the ETW layer ofthe telemetry client.
- **EventsPersistedCount** NumberEventSubStoreResetCounter The number of events that reached the PersistEvent stage.
- **EventStoreLifetimeResetCounter** Number of times event DB was reset for the lifetime of UTC.
- **EventStoreResetCounter** Number of times event DBdatabase was reset.
- **EventStoreResetSizeSum** TotalEventSubStoreResetSizeSum The total size ofthe event DBdatabase across all resets reports in this instance.
- **EventsUploaded** NumberThe number of eventsthat have been uploaded.
- **Flags** Flags indicatingthat indicate device state, such as network state, battery state, and opt-in state.
- **FullTriggerBufferDroppedCount** NumberThe number of eventsthat were dropped due tobecause the trigger buffer beingwasfull.
- **HeartBeatSequenceNumber** The sequence number of thisA monotonically increasing heartbeatcounter.
- **InvalidHttpCodeCount** NumberThe number of invalid HTTP codes received from contacting Vortex.
- **LastAgentConnectionError** LastThe last non-timeout error encounteredthat happened in the host/agent channel.
- **LastEventSizeOffender** EventThe name ofthe last event whichthat exceeded maxthe maximum event size.
- **LastInvalidHttpCode** LastThe last invalid HTTP code received from Vortex.
- **MaxActiveAgentConnectionCount** The maximum number of active agents during this heartbeat timeframe.
- **MaxInUseScenarioCounter** SoftThe soft maximum number of scenarios loaded by UTCthe Connected User Experience and Telemetry component.
- **PreviousHeartBeatTime** TimeThe time of last heartbeat event (. This allows chaining of events).
- **RepeatedUploadFailureDropped** Number of events lost due to repeated upload failures for a single buffer.
- **SettingsHttpAttempts** NumberThe number of attempts to contactthe OneSettings service.
- **SettingsHttpFailures** The number of failures from contacting the OneSettings service.
- **ThrottledDroppedCount** NumberThe number of events dropped due to throttling of noisy providers.
- **TopUploaderErrors** List of top errors received from the upload endpoint.
- **UploaderDroppedCount** NumberThe number of events dropped atby the uploader layer ofthe telemetry client.
- **UploaderErrorCount** Number of errors received from the upload endpoint.
- **VortexFailuresTimeout** The number of timeout failures received from Vortex.
- **VortexHttpAttempts** NumberThe number of attempts to contactthe Vortexservice.
- **VortexHttpFailures4xx** NumberThe number of 400-499 error codes received from Vortex.
- **VortexHttpFailures5xx** NumberThe number of 500-599 error codes received from Vortex.
- **VortexHttpResponseFailures** Number of Vortex responses that are not 2XX or 400.
- **VortexHttpResponsesWithDroppedEvents** Number of Vortex responses containing at least 1 dropped event.

TelClientSyntheticDxgKernelTelemetry events DxgKrnlTelemetry.HeartBeat_Aria_5GPUAdapterInventoryV2

This
event is the telemetry client ARIA heartbeatsends basic GPU and display driver information to keep Windows and display drivers up-to-date.

The following fields are available:

- **CompressedBytesUploaded** Number of compressed bytes uploadedaiSeqId The event sequence ID.
- **CriticalDataDbDroppedCount** Number of critical data sampled events dropped at the database layerbootId The system boot ID
.
- **CriticalOverflowEntersCounter** Number of times critical overflow mode was entered in eventdatabaseComputePreemptionLevel The maximum preemption level supported by GPU for compute payload.
- **DbCriticalDroppedCount** Total numberDedicatedSystemMemoryB The amount of dropped critical eventssystem memory dedicated for GPU use (in event database.
- **DbDroppedCount** Numberbytes). DedicatedVideoMemoryB The amount of events dropped at the database layer.

- **DbDroppedFailureCount** Numberdedicated VRAM of events dropped due to database failures.
- **DbDroppedFullCount** Number of events dropped due to database being full.
- **EnteringCriticalOverflowDroppedCounter** Number of events dropped due to critical overflow mode being initiated.
- **EventsPersistedCount** Number of events that reached the PersistEvent stage.
- **EventStoreLifetimeResetCounter** Number of times the event store has been reset.
- **EventStoreResetCounter** Number of times the event store has been reset during this heartbeat.
- **EventStoreResetSizeSum** Size of event store resetGPU (in bytes.
- **EventsUploaded** Number of events uploaded.
- **HeartBeatSequenceNumber). DisplayAdapterLuid** The sequence number of this heartbeatdisplay adapter LUID.
- **InvalidHttpCodeCount** Number of invalid HTTP codes received from contacting Vortex.
- **LastEventSizeOffender** Event name of last event which exceeded max event size.
- **LastInvalidHttpCode** Last invalid HTTP code received from Vortex.
- **PreviousHeartBeatTimeDriverDate** The FILETIMEdate of the previous heartbeat firedisplay driver.
- **RepeatedUploadFailureDropped** NumberDriverRank The rank of events lost due to repeated upload failures for a singlebuffer.
- **SettingsHttpAttempts** Number of attempts to contact OneSettings service.
- **SettingsHttpFailures** Number of failures from contacting OneSettings service.
- **TopUploaderErrors** List of top errors received from the upload endpointdisplay driver.
- **UploaderDroppedCount** Number of events dropped at the uploader layer of telemetry clientDriverVersion The display driver version.
- **UploaderErrorCount** Number of errors received from the upload endpointGPUDeviceID The GPU device ID.
- **VortexFailuresTimeout** Number of time out failures received from Vortex.
- **VortexHttpAttempts** Number of attempts to contact Vortex.
- **VortexHttpFailures4xx** Number of 400-499 error codes received from Vortex.
- **VortexHttpFailures5xx** Number of 500-599 error codes received from Vortex.
- **VortexHttpResponseFailures** Number of Vortex responses that are not 2XX or 400.
- **VortexHttpResponsesWithDroppedEvents** Number of Vortex responses containing at least 1 dropped event.

TelClientSynthetic.HeartBeat_Seville_5

This event is sentGPUPreemptionLevel The maximum preemption level supported by the universal telemetry client (UTC) as aheartbeat signalGPU for Sensegraphics payload.

GPURevisionID The following fields are available:

- **AgentConnectionErrorsCount** Number of non-timeout errors associated with the host or agent channelGPU revision ID.
- **CompressedBytesUploaded** Number of compressed bytes uploadedGPUVendorID The GPU vendor ID.
- **ConsumerDroppedCount** Number of events dropped at consumer layer of the telemetry clientInterfaceId The GPU interface ID.
- **CriticalDataDbDroppedCount** Number of critical data sampled events dropped atIsDisplayDevice Does the database layer.
- **CriticalDataThrottleDroppedCount** Number of critical data sampled events dropped due to throttling.
- **CriticalOverflowEntersCounter** Number of times critical overflow mode was enteredGPU have displaying capabilities? IsHybrid Discrete Does the GPU have discrete GPU capabilities in event database.
- **DailyUploadQuotaInBytes** Daily upload quota for Sensea hybrid device? IsHybridIntegrated Does the GPU have integrated GPU capabilities in bytes (only in in-proc mode).
- **DbCriticalDroppedCount** Total numbera hybrid device? IsLDA Is the GPU comprised of dropped critical events in eventdatabase.
- **DbDroppedCount** Number of events dropped due to database being full.
- **DbDroppedFailureCount** Number of events dropped due to database failures.
- **DbDroppedFullCount** Number of events dropped due to database being full.
- **DecodingDroppedCount** Number of events dropped due to decoding failures.
- **DiskSizeInBytes** Size of event store for Sense in bytes (only in in-proc mode).
- **EnteringCriticalOverflowDroppedCounter** Number of events dropped due to critical overflow mode being initiated.
- **EtwDroppedBufferCount** Number of buffers dropped inLinked Display Adapters? IsMiracastSupported Does the universaltelemetry client (UTC) event tracing for Windows (ETW) session.
- **EtwDroppedCount** Number of events droppedGPU support Miracast? IsMismatchLDA Is atleast one device in the event tracingfor Windows (ETW) layer of telemetry client.

- **EventsPersistedCount** Number of events that reached~~Linked Display Adapters chain from a different vendor? IsMPOSupported Does~~ the PersistEvent stage.
- **EventStoreLifetimeResetCounter** Number of times event~~GPU support Multi-Plane Overlays? IsMsMiracastSupported Are~~ the database was reset for~~GPU Miracast capabilities driven by a Microsoft solution? IsPostAdapter Is this GPU~~ the lifetime of~~POST GPU in~~ the universal telemetry client (UTC).
- **EventStoreResetCounter** Number of times~~device? IsRenderDevice Does~~ the event database was reset.
- **EventStoreResetSizeSum** Total size~~GPU have rendering capabilities? IsSoftwareDevice Is this a software implementation~~ of the event database across all resets reports in this instance.
- **EventsUploaded** Number of events uploaded.
- **Flags** Flags indicating~~GPU? MeasureEnabled Is the~~ device state, such as network state, battery state, and opt-in state.
- **FullTriggerBufferDroppedCount** Number of events dropped due~~listening~~ to trigger buffer being full.
- **HeartBeatSequenceNumber**~~**MICROSOFT_KEYWORD_MEASURES? NumVidPnSources**~~ The sequence number of this heartbeat~~supported display output sources~~.
- **InvalidHttpCodeCount** Number of invalid HTTP codes received from contacting Vortex.
- **LastAgentConnectionError** Last non-timeout error encountered in the host/agent channel.
- **LastEventSizeOffender** Event name of last event which exceeded the maximum event size.
- **LastInvalidHttpCode** Last invalid HTTP code received from Vortex.
- **MaxActiveAgentConnectionCount** Maximum~~NumVidPnTargets The~~ number of active agents during this heartbeat timeframe~~supported display output targets~~.
- **NormalUploadTimerMillis** Number~~SharedSystemMemoryB The amount~~ of milliseconds between each upload of normal eventsf or SENSE~~system memory shared by GPU and CPU~~ (only in in-proc mode~~bytes~~).
- **PreviousHeartBeatTime** Time of last heartbeat event (allows chaining of events).
- **RepeatedUploadFailureDropped** Number of events lost due to repeated failed uploaded attempts~~SubSystemID The subsystem ID~~.
- **SettingsHttpAttempts** Number of attempts to contact OneSettings service~~SubVendorID The GPU sub-vendor ID~~.
- **SettingsHttpFailures** Number of failures from contacting~~TelemetryEnabled Is~~ the OneSettings service.
- **ThrottledDroppedCount** Number of events dropped due~~device listening~~ to throttling of noisy providers.
- **TopUploaderErrors** Top uploader errors, grouped by endpoint and error type.
- **UploaderDroppedCount** Number of events dropped at the uploader layer of the telemetry client.
- **UploaderErrorCount** Number of input for the TopUploaderErrors mode estimation.
- **VortexFailuresTimeout** Number of time out failures received from Vortex.
- **VortexHttpAttempts** Number of attempts~~MICROSOFT_KEYWORD_TELEMETRY? TelInvEvntTrigger What triggered this event~~ to contact Vortex.
- **VortexHttpFailures4xx** Number of 400-499 error codes received from Vortex.
- **VortexHttpFailures5xx** Number of 500-599 error codes received from Vortex.
- **VortexHttpResponseFailures** Number of Vortex responses that are not 2XX~~be logged? Example: 0 (GPU enumeration)~~ or 400.
- **VortexHttpResponsesWithDroppedEvents** Number of Vortex responses containing at least 1 dropped~~(DxgKrnlTelemetry provider toggling) version The~~ event~~version.WDDMVersion The Windows Display Driver Model version.~~

Direct to update~~Fault Reporting~~ events

Microsoft.Windows.DirectToUpdate~~FaultReporting~~.DTUCoordinatorCheckApplicabilityGenericFailure~~AppCrashEvent~~

This
event indicatse that we have received an unexpected error in the Direct~~sends data about crashes for both native and managed applications,~~ to Update (DTU) Coordinators CheckApplicability call~~help keep Windows up to date~~.

The following fields are available:

- **CampaignID** ID of~~data includes information about~~ the campaign being run.
- **ClientID** ID~~crashing process and a summary~~ of the client receiving the update~~its exception record~~.
- **CoordinatorVersion** Coordinator version of Direct to Update~~It does not contain any Watson bucketing information~~.
- **CV** Correlation vector.
- **hResult** HRESULT of the failure.

Microsoft.~~The bucketing information is recorded in a~~ Windows.DirectToUpdate.DTUCoordinatorCleanupGenericFailure

This~~Error Reporting (WER)~~ event indicates that we have received an unexpected error in~~is generated when~~ the Direct~~WER client reports the crash~~ to Update (DTU) Coordinator Cleanup call.

The following fields are available:

- **CampaignID** Campaign ID being run
- **ClientID** Client ID being run
- **CoordinatorVersion** Coordinator version of DTU
- **CV** Correlation vector
- **hResult** HRESULT of the failure

Microsoft.Windows.DirectToUpdate.DTUCoordinatorCleanupSuccess

This event indicates that~~Watson service, and~~ the Coordinator Cleanup call succeeded.

The following fields are available:

- **CampaignID** Campaign ID being run
- **ClientID** Client ID being run
- **CoordinatorVersion** Coordinator version of DTU
- **CV** Correlation vector

Microsoft.Windows.DirectToUpdate.DTUCoordinatorCommitGenericFailure

This~~WER~~ event indicates that we have received an unexpected error in~~will contain~~ the Direct to Update~~same ReportID~~ (DTU)Coordinator Commit call.

The following fields are available:

- **CampaignID** Campaign ID being run.
- **ClientID** Client ID being run.
- **CoordinatorVersion** Coordinator version~~see field 14~~ of DTU.
- **CV** Correlation vector.
- **hResult** HRESULT~~crash event, field 19~~ of~~WER event) as~~ the failure.

Microsoft.Windows.DirectToUpdate.DTUCoordinatorCommitSuccess

This~~crash~~ event indicates that~~for~~ the Coordinator Commit call succeeded.

The following fields are available:

- **CampaignID** Campaign ID~~crash~~ being run~~reported~~.
- **ClientID** Client ID being run~~AppCrash is emitted once for each crash handled by WER (~~e.
- **CoordinatorVersion** Coordinator version of DTU~~g~~.
- **CV** Correlation vector.

Microsoft.Windows.DirectToUpdate.DTUCoordinatorDownloadGenericFailure

This~~from an unhandled exception or FailFast or ReportException). Note that Generic Watson~~ event indicates that we have receivedan un expected error in the Direct to Update~~types~~ (DTU) Coordinator Download call~~e~~.

The following fields are available:

- **CampaignID** Campaign ID being run~~g~~.
- **ClientID** Client ID being run.
- **CoordinatorVersion** Coordinator version of DTU.
- **CV** Correlation vector.
- **hResult** HRESULT of the failure.

Microsoft.Windows.DirectToUpdate.DTUCoordinatorDownloadIgnoredFailure

This event indicates that we have received an error in the Direct to Update (DTU~~from PLM~~) Coordinator Download call that will~~may~~be ignored~~considered crashes" by a user DO NOT emit this event~~.

The following fields are available:

- **CampaignID** Campaign ID being run.
- **ClientID** Client ID being run.
- **CoordinatorVersion** Coordinator version~~AppName The name~~ of DTU.
- **CV** Correlation vector.
- **hResult** HRESULT of the failure.

Microsoft.Windows.DirectToUpdate.DTUCoordinatorDownloadSuccess

This event indicates~~app~~ that the Coordinator Download call succeeded~~has crashed~~.

The following fields are available:

- **CampaignID** Campaign~~AppSessionGuid GUID made up of process~~ ID being run.
- **ClientID** Client ID being run.
- **CoordinatorVersion** Coordinator version of DTU.
- **CV** Correlation~~and is used as a correlation~~ vector.

Microsoft.Windows.DirectToUpdate.DTUCoordinatorHandleShutdownGenericFailure

This event indicates that we have received an unexpected error~~for process instances~~ in the Direct to Update (DTU) CoordinatorHandleShutdown call~~telemetry backend~~.

~~AppTimeStamp~~ The following fields are available:

- **CampaignID** Campaign ID being run~~date/time stamp of the app~~.
- **ClientID** Client ID being run.
- **CoordinatorVersion** Coordinate~~AppVersion The~~ version of DTU.
- **CV** Correlation vector.
- **hResult** HRESULT of the failure.

Microsoft.Windows.DirectToUpdate.DTUCoordinatorHandleShutdownSuccess

This event indicates~~app~~ that the Coordinator HandleShutdown call succeeded~~has crashed~~.

~~ExceptionCode~~ The following fields are available:

- **CampaignID** Campaign ID being run.
- **ClientID** Client ID being run.
- **CoordinatorVersion** Coordinator version of DTU.

- **CV** Correlation vector.

Microsoft.Windows.DirectToUpdate.DTUCoordinatorInitializeGenericFailure

This event indicates~~exception code returned by the process~~ that we have received an unexpected error in the Direct to Update(DTU) Co ordinator Initialize call~~has crashed~~.

~~ExceptionOffset~~ The following fields are available:

- **CampaignID** Campaign ID being run~~address where the exception had occurred~~.
- **ClientID** Client ID being run~~Flags Flags indicating how reporting is done~~.
- **CoordinatorVersion** Coordinator version of DTU.
- **CV** Correlation vector.
- **hResult** HRESULT of~~For example, queue~~ the failure~~report, do not offer JIT debugging, or do not terminate the process after repo rting~~.

Microsoft~~ModName Exception module name (e~~.Windows~~g~~.DirectToUpdate~~bar~~.DTUCoordinatorInitializeSuccess

This event indicates that~~dll). ModTimeStamp The date/time stamp of~~ the Coordinator Initialize call succeeded~~module~~.

~~ModVersion~~ The following fields are available:

- **CampaignID** Campaign ID being run.
- **ClientID** Client ID being run.
- **CoordinatorVersion** Coordinator version of DTU.
- **CV** Correlation vector.

Microsoft.Windows.DirectToUpdate.DTUCoordinatorInstallGenericFailure

This event indicates~~the module~~ that we have received an unexpected error in the Direct to Update (DTU) Coordinator Install call~~has cras hed~~.

The following fields are available:

- **CampaignID** Campaign ID being run~~PackageFullName Store application identity~~.
- **ClientID** Client ID being run~~PackageRelativeAppId Store application identity~~.
- **CoordinatorVersion** Coordinator version~~ProcessArchitecture Architecture~~ of DTU.
- **CV** Correlation vector.
- **hResult** HRESULT~~the crashing process, as one~~ of the failure~~PROCESSOR_ARCHITECTURE_* constants: 0: PROCESSOR_ARCHITECTURE_INTEL~~.

Microsoft~~5: PROCESSOR_ARCHITECTURE_ARM~~.Windows~~9: PROCESSOR_ARCHITECTURE_AMD64~~.DirectToUpdate~~12 : PROCESSOR_ARCHITECTURE_ARM64~~.DTUCoordinatorInstallIgnoredFailure

This event indicates that we have received an error in the Direct to Update (DTU) Coordinator Install call that will be ignored.

~~ProcessCreateTime~~ The following fields are available:

- **CampaignID** Campaign ID being run.
- **ClientID** Client ID being run.
- **CoordinatorVersion** Coordinator version~~time~~ of DTU.
- **CV** Correlation vector.
- **hResult** HRESULT~~creation~~ of the failure.

Microsoft.Windows.DirectToUpdate.DTUCoordinatorInstallSuccess

This event indicates~~process~~ that the Coordinator Install call succeeded~~has crashed~~.

~~ProcessId~~ The following fields are available:

- **CampaignID** Campaign ID being run.
- **ClientID** Client ID being run.
- **CoordinatorVersion** Coordinator version of DTU.
- **CV** Correlation vector.

Microsoft.Windows.DirectToUpdate.DTUCoordinatorProgressCallBack

This event indicates~~the process~~ that the Coordinator's progress callback has been called~~crashed~~.

The following fields are available:

- **CampaignID** Campaign ID being run~~ReportId A GUID used to identify the report~~.
- **ClientID** Client ID being run.
- **CoordinatorVersion** Coordinator version of DTU.
- **CV** Correlation vector.
- **DeployPhase** Current Deploy Phase.

Microsoft.Windows.DirectToUpdate.DTUCoordinatorSetCommitReadySuccess

This event indicates that~~can used to track~~ the Coordinator SetCommitReady call succeeded~~report across Watson~~.

~~TargetAppId~~ The following fields are available:

- **CampaignID** ID~~kernel reported AppId~~ of the update campaign~~application~~ being run~~reported~~.
- **ClientID** ID~~TargetAppVer The specific version~~ of the client receiving~~application being reported TargetAsId The sequence number for~~ the update~~hanging process~~.
- **CoordinatorVersion** Coordinator version of Direct to Update.
- **CV** Correlation vector.

Microsoft.Windows.DirectToUpdate.DTUCoordinatorWaitForRebootUiNotShown

This event indicates that the Coordinator WaitForRebootUi call succeeded.

The following fields are available:

- **CampaignID** Campaign ID being run.
- **ClientID** ID of the client receiving the~~Feature~~ update.
- **CoordinatorVersion** Coordinator version of Direct to Update.
- **CV** Correlation vector.
- **hResult** HRESULT of the failure.~~events~~

Microsoft.Windows.DirectToUpdate~~Upgrade~~.DTUCoordinatorWaitForRebootUiSelection~~Uninstall.UninstallGoBack ButtonClicked~~

This event indicates that~~sends basic metadata about~~ the user selected an option on~~starting point of uninstalling a feature update, which helps ensure customers can safely revert to a well-known state if~~ the Reboot UI~~update caused any problems~~.

The following fields are available:

- **CampaignID** ID of the update campaign being run.
- **ClientID** ID of the client receiving the update.
- **CoordinatorVersion** Coordinator version of Direct to Update.
- **CV** Correlation vector.
- **rebootUiSelection** Selection on the Reboot UI.~~Hang Reporting events~~

Microsoft.Windows.DirectToUpdate~~HangReporting~~.DTUCoordinatorWaitForRebootUiSuccess~~AppHangEvent~~

This event indicates that the Coordinator WaitForRebootUi call succeeded.

The following fields are available:

- **CampaignID** ID of the update campaign being run.
- **ClientID** ID of the client receiving the update.
- **CoordinatorVersion** Coordinator version of Direct~~sends data about hangs for both native and managed applications,~~ to Update.
- **CV** Correlation vector.

Microsoft.~~help keep~~ Windows~~up to date~~.DirectToUpdate~~It does not contain any Watson bucketing information~~.DTUHandlerCheckApplicabilityInternalGenericFailure

This event indicates that we have received an unexpected error~~The bucketing information is recorded~~ in the Direct to Update~~a Windows Error Reporting~~ (DTU~~WER~~) Handler CheckApplicabilityInternal call.

The following fields are available:

- **CampaignID** ID of~~event that is generated when~~ the campaign being run.
- **ClientID** ID of the~~WER~~ client receiving~~reports~~ the update.
- **CoordinatorVersion** Coordinator version of Direct~~hang~~ to Update.
- **CV** Correlation vector.
- **hResult** HRESULT of the failure.

Microsoft.Windows.DirectToUpdate.DTUHandlerCheckApplicabilityInternalSuccess

This~~Watson service, and the WER~~ event indicates that~~will contain~~ the Handler CheckApplicabilityInternal call succeeded.

The following fields are available:

- **ApplicabilityResult** The result~~same ReportID (see field 13~~ of the applicability check.
- **CampaignID** ID~~hang event, field 19~~ of~~WER event) as~~ the update campaign~~hang event for the hang~~ being run~~reported~~.
- **ClientID** ID of the client receiving the update~~AppHang is reported only on PC devices~~.
- **CoordinatorVersion** Coordinator version of Direct to Update~~It handles classic Win32 hangs and is emitted only once per report~~.
- **CV** Correlation vector~~Some behaviors that may be perceived by a user as a hang are reported by app managers (e~~.

Microsoft~~g~~.Windows~~PLM/RM/EM) as Watson Generics and will not produce AppHang events~~.DirectToUpdate.DTUHandlerCheckApplicabilitySuccess

This event indicates that the Handler CheckApplicability call succeeded.

The following fields are available:

- **ApplicabilityResult~~AppName~~** The result code indicating whether the update is applicable.
- **CampaignID** ID~~name~~ of the update campaign being run~~app that has hung~~.

- **ClientID** IDAppSessionGuid GUID made up of the client receiving the update.
- **CoordinatorVersion** Coordinator version of Direct to Update.
- **CV** Correlation vector.
- **CV_new** Newprocess id used as a correlation vector.

Microsoft.Windows.DirectToUpdate.DTUHandlerCheckIfCoordinatorMinApplicableVersionSuccess

This event indicates thatfor process instances in the Handler CheckIfCoordinatorMinApplicableVersion call succeededtelemetry backend.

AppVersion The following fields are available:

- **CampaignID** IDversion of the update campaign being runapp that has hung.
- **CheckIfCoordinatorMinApplicableVersionResult** Result of CheckIfCoordinatorMinApplicableVersion functionPackageFullName Store application identity.
- **ClientID** IDPackageRelativeAppId Store application identity. ProcessArchitecture Architecture of the client receiving the update.
- **CoordinatorVersion** Coordinator versionhung process, as one of Direct to Update.
- **CV** Correlation vector.

Microsoft.Windows.DirectToUpdate.DTUHandlerCommitGenericFailure

This event indicates that we have received an unexpected error in the Direct to Update (DTU) Handler CommitcallPROCESSOR_ARCHITECTURE_* constants: 0: PROCESSOR_ARCHITECTURE_INTEL.

The following fields are available5:

- **CampaignID** ID of the update campaign being runPROCESSOR_ARCHITECTURE_ARM.
- **ClientID** ID of the client receiving the update9: PROCESSOR_ARCHITECTURE_AMD64.
- **CoordinatorVersion** Coordinator version of Direct to Update.
- **CV** Correlation vector.
- **CV12: PROCESSOR_new** New correlation vectorARCHITECTURE_ARM64.
- **hResult** HRESULTProcessCreateTime The time ofcreation of the failure.

Microsoft.Windows.DirectToUpdate.DTUHandlerCommitSuccess

This event indicatesprocess that the Handler Commit call succeededhas hung.

ProcessId The following fields are available:

- **CampaignID** ID of the update campaign being runprocess that has hung.run
- **ClientID** ID ofReportId A GUID used to identify the client receiving the updatereport.
- **CoordinatorVersion** Coordinator version of DirectThis can used to Update.
- **CV** Correlation vector.
- **CV_new** New correlation vector.

Microsoft.Windows.DirectToUpdate.DTUHandlerDownloadAndExtractCabFailure

This event indicates thattrack the Handler Download and Extract cab call failedreport across Watson.

TargetAppId The following fields are available:

- **CampaignID** IDkernel reported AppId of the update campaignapplication being runreported.
- **ClientID** IDTargetAppVer The specific version of the client receiving the updateapplication being reported.
- **CoordinatorVersion** Coordinator version of Direct to Update.
- **CV** Correlation vector.

- **DownloadAndExtractCabFunction_failureReason** Reason whyTargetAsId The sequence number for the update download andextracthanging process failed.
- **hResult** HRESULT ofTypeCode Bitmap describing the failurehang type.

Microsoft.Windows.DirectToUpdate.DTUHandlerDownloadAndExtractCabSuccess

This event indicates thatWaitingOnAppName If this is a cross process hang waiting for an application, this has the HandlerDownload and Extract cab call succeeded.

The following fields are available:

- **CampaignID** IDname of the update campaign being runapplication.
- **ClientID** ID ofWaitingOnAppVersion If this is a cross process hang, this has the client receiving the update.
- **CoordinatorVersion** Coordinator version of Direct to Updatethe application for which it is waiting.
- **CV** Correlation vector.

Microsoft.Windows.DirectToUpdate.DTUHandlerDownloadGenericFailure

This event indicates that we have received an unexpected error inWaitingOnPackageFullName If this is a cross process hang waiting for a package, this has the Direct to Update (DTU) Handler Download call.

The following fields are available:

- **CampaignID** IDfull name of the update campaign being runpackage for which it is waiting.
- **ClientID** ID ofWaitingOnPackageRelativeAppId If this is a cross process hang waiting for a package, this has the client receivingthe update.
- **CoordinatorVersion** Coordinator versionrelative application id of Direct to Update.
- **CV** Correlation vector.
- **hResult** HRESULT of the failurepackage.

Microsoft.Windows.DirectToUpdate.DTUHandlerDownloadSuccessInventory events

This event indicates that the Handler Download call succeeded.

The following fields are available:

- **CampaignID** ID of the update campaign being run.
- **ClientID** ID of the client receiving the update.
- **CoordinatorVersion** Coordinator version of Direct to Update.
- **CV** Correlation vector.ChecksumDictionary

Microsoft.Windows.DirectToUpdate.DTUHandlerInitializeGenericFailure

This event indicates that we have received an unexpected error in the Direct to Update (DTU) Handler Initialize callThe list of values sent by each object type.

The following fields are available:

- **CampaignID** ID of the update campaignKey The object type being rundescribed.
- **ClientID** IDValue The number of the client receiving the update.
- **CoordinatorVersion** Coordinator versionobjects of Direct to Updatethis type that were sent.
- **CV** Correlation vector.
- **DownloadAndExtractCabFunction_hResult** HRESULT of the download and extract.
- **hResult** HRESULT of the failure.

Microsoft.Windows.DirectToUpdate.DTUHandlerInitializeSuccessCOMPID

This event indicatesprovides a device's internal application compatible ID, a vendor-defined identification that the Handler Initializecall succeededWindows uses to match a device to an INF file.A device can have a list of compatible IDs associated with it.

The following fields are available:

- **CampaignID** IDOrder The index of the update campaign being run.
- **ClientID** IDarray ofcompatible IDs for the client receiving the updatedevice.
- **CoordinatorVersion** Coordinator versionValue The array of Direct to Update.
- **CV** Correlation vector.
- **DownloadAndExtractCabFunction_hResult** HRESULT ofcompatible IDs for the download and extractiondevice.

Microsoft.Windows.DirectToUpdate.DTUHandlerInstallGenericFailureHWID

This event indicatesprovides a device's internal hardware ID, a vendor-defined identification that we have receivedWindows uses to match a device to an unexpected error in the Direct to Update (DTU) Handler Install callINF file.In most cases, a device has associated with it a list of hardware IDs.

The following fields are available:

- **CampaignID** IDOrder The index of the update campaign being run.
- **ClientID** IDarray ofinternal hardware IDs for the client receiving the updatedevice.
- **CoordinatorVersion** Coordinator versionValue The array of Direct to Update.
- **CV** Correlation vector.
- **hResult** HRESULT ofinternal hardware IDs for the failuredevice.

Microsoft.Windows.DirectToUpdate.DTUHandlerInstallSuccessInstallDateArpLastModified

This event
indicates that the Coordinator Install call succeededdate the add/remove program (ARP) entry was last modified by an update.

The following fields are available:

- **CampaignID** IDOrder The index of the update campaign being runordered array.
- **ClientID** ID ofValue The value contained in the client receiving the updateordered array.
- **CoordinatorVersion** Coordinator version of Direct to Update.
- **CV** Correlation vector.

Microsoft.Windows.DirectToUpdate.DTUHandlerSetCommitReadySuccessInstallDateFromLinkFile

This event indicates thatprovides the Handler SetCommitReady call succeededapplication installation date from the linked file.

The following fields are available:

- **CampaignID** IDOrder The index of the campaign being runordered array.
- **ClientID** ID ofValue The value contained in the client receiving the updateordered array.
- **CoordinatorVersion** Coordinator version of Direct to Update.
- **CV** Correlation vector.

Microsoft.Windows.DirectToUpdate.DTUHandlerWaitForRebootUiGenericFailureInstallDateMsi

This event indicates that we have received an unexpected error in The install date from the Direct to Update Microsoft installer (DTU MSI) Handler WaitForRebootUi call database.

The following fields are available:

- **CampaignID Order** The ID index of the campaigning being run ordered array.
- **ClientID** ID of the client receiving the update.
- **CoordinatorVersion** Coordinator version of Direct to Update.
- **CV** Correlation vector.
- **hResult Value** The HRESULT of value contained in the failure ordered array.

Microsoft.Windows.DirectToUpdate.DTUHandlerWaitForRebootUiSuccess

This event indicates that the Handler WaitForRebootUi call succeeded.

The following fields are available:

- **CampaignID** ID of the campaign being run.
- **ClientID** ID of the client receiving the update.
- **CoordinatorVersion** Coordinator version of Direct to Update.
- **CV** Correlation vector.

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:

- **DeviceCensus Device** A count of device census objects in cache.
- **DriverPackageExtended DeviceCensus** A count of driverpackageextended devicecensus objects in cache.
- **FileSigningInfo DriverPackageExtended** A count of file signing driverpackageextended objects in cache.
- **InventoryApplicationFile** A count of application file objects in cache.
- **InventoryApplicationAppVFileSigningInfo** A count of application AppV file signing objects in cache.
- **InventoryApplicationDriverGeneric** A count of application driver generic objects in cache
- **InventoryApplicationFile. HwItem** A count of application file hwitem objects in cache.
- **InventoryApplicationFrameworkInventoryApplication** A count of application framework objects in cache
- **InventoryApplicationShortcut. InventoryApplicationFile** A count of application shortcut file objects in cache.
- **InventoryDeviceContainer** A count of device container objects in cache.
- **InventoryDeviceInterface** A count of Plug and Play device interface objects in cache.
- **InventoryDeviceMediaClass** A count of device media objects in cache.
- **InventoryDevicePnp** A count of device Plug and Play objects in cache.
- **InventoryDeviceUsbHubClass** A count of device usb objects in cache
- **InventoryDriverBinary** A count of driver binary objects in cache.
- **InventoryDriverPackage** A count of device objects in cache.
- **InventoryMiscellaneousOfficeAddInMetadata** A count of office add-in metadata objects in cache
- **InventoryMiscellaneousOfficeAddInUsage. Orphan** A count of office add-in usage orphan file objects in cache.
- **InventoryMiscellaneousOfficeIdentifiersPrograms** A count of office identifier program objects in cache
- **InventoryMiscellaneousOfficeIESettings** A count of office ie settings objects in cache
- **InventoryMiscellaneousOfficeInsights** A count of office insights objects in cache
- **InventoryMiscellaneousOfficeProducts** A count of office products objects in cache
- **InventoryMiscellaneousOfficeSettings** A count of office settings objects in cache
- **InventoryMiscellaneousOfficeVBA** A count of office vba objects in cache

- **InventoryMiscellaneousOfficeVBARuleViolations** A count of office vba rule violations objects in cache
- **InventoryMiscellaneousUUPInfo** A count of uup info objects in cache

Microsoft.Windows.Inventory.Core.AmiTelCacheFileInfo

Diagnostic data about the inventory cache.

The following fields are available:

- **CacheFileSize** Size of the cache.
- **InventoryVersion** Inventory version of the cache.
- **TempCacheCount** Number of temp caches created.
- **TempCacheDeletedCount** Number of temp caches deleted.

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

Microsoft.Windows.Inventory.Core.InventoryApplicationAdd

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

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **HiddenArp** Indicates whether a program hides itself from showing up in ARP.
- **InstallDate** The date the application was installed (a best guess based on folder creation date heuristics).
- **InstallDateArpLastModified** The date of the registry ARP key for a given application. Hints at install date but not always accurate. Passed as an array. Example: 4/11/2015 00:00:00~~See InstallDateArpLastModified .~~
- **InstallDateFromLinkFile** The estimated date of install based on the links to the files. Passed as an array.~~See InstallDateFromLinkFile .~~
- **InstallDateMsi** The install date if the application was installed via Microsoft Installer (MSI). Passed as an array.~~See InstallDateMsi .~~
- **InventoryVersion** The version of the inventory file generating the events.
- **Language** The language code of the program.
- **MsiPackageCode** A GUID that describes the MSI Package. Multiple 'Products' (apps) can make up an MsiPackage.
- **MsiProductCode** A GUID that describe the MSI Product.
- **Name** The name of the application.
- **OSVersionAtInstallTime** The four octets from the OS version at the time of the application's install.
- **PackageFullName** The package full name for a Store application.
- **ProgramInstanceId** A hash of the file IDs in an app.
- **Publisher** The Publisher of the application. Location pulled from depends on the 'Source' field.
- **RootDirPath** The path to the root directory where the program was installed.
- **Source** How the program was installed (for example, ARP, MSI, Appx).
- **StoreAppType** A sub-classification for the type of Microsoft Store app, such as UWP or Win8StoreApp.
- **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.
- **Version** The version number of the program.

Microsoft.Windows.Inventory.Core.InventoryApplicationDriverAdd

This event represents what drivers an application installs.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **InventoryVersion** The version of the inventory component
- **ProgramIds** The unique program identifier the driver is associated with

Microsoft.Windows.Inventory.Core.InventoryApplicationDriverStartSync

The InventoryApplicationDriverStartSync event indicates that a new set of InventoryApplicationDriverStartAdd events will be sent.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **InventoryVersion** The version of the inventory component.

Microsoft.Windows.Inventory.Core.InventoryApplicationFrameworkAdd

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

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

Microsoft.Windows.Inventory.Core.InventoryApplicationFrameworkStartSync

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

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

Microsoft.Windows.Inventory.Core.InventoryApplicationRemove

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

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **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 Plug and Play device) to help keep Windows up to date.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

Microsoft.Windows.Inventory.Core.InventoryDeviceContainerRemove

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

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **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.
- **EnergyMeter** Indicates if an Energy 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.
- **InventoryVersion** The version of the inventory file generating the events.
- **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.

Microsoft.Windows.Inventory.Core.InventoryDeviceInterfaceStartSync

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

This event includes fields from Ms.Device.DeviceInventoryChange.

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 Plug and Play device that is specific to a particular class of devices to help keep Windows up to date while reducing overall size of data payload.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **Audio_CaptureDriver** The Audio device capture driver endpoint.
- **Audio_RenderDriver** The Audio device render driver endpoint.
- **InventoryVersion** The version of the inventory file generating the events.

Microsoft.Windows.Inventory.Core.InventoryDeviceMediaClassRemove

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

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

Microsoft.Windows.Inventory.Core.InventoryDevicePnpAdd

This event represents the basic metadata about a plug and play (PNP) device and its associated driver.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **BusReportedDescription** The description of the device reported by the bux.
- **Class** The device setup class of the driver loaded for the device.
- **ClassGuid** The device class GUID from the driver package
- **COMPID** The device setup class guid~~A JSON array the provides the value and order~~ of the driver loaded~~compatible ID tree~~ for the device.~~See COMPID .~~
- **ContainerId** The list of compat ids for the device.
- **Description** System-supplied~~A system-supplied~~ GUID that uniquely groups the functional devices associated with a single-function or multifunction device installed in the computer~~device~~.
- **DeviceState~~Description~~** ~~The~~ device description.
- **DriverId** 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
- **DriverName~~DriverId~~** ~~A~~ unique identifier for the driver installed~~device~~.
- **DriverPackageStrongName~~DriverName~~** ~~The~~ immediate parent directory name in the Directory field of InventoryDriverPackage
- **DriverVerDate** Name of the .sys~~driver~~ image file (or wudfrd.sys if using user mode driver framework).
- **DriverVerVersion~~DriverVerDate~~** ~~The~~ immediate parent directory name in the Directory field of InventoryDriverPackage.
- **Enumerator** The date of the driver loaded for the device.
- **HWID~~DriverVerVersion~~** ~~The~~ version of the driver loaded for the device.
- **Inf~~Enumerator~~** ~~The~~ bus that enumerated the device~~HWID A JSON array that provides the value and order of the HWID tree for the device.See HWID . Inf The INF file name.~~
- **InstallState** The device installation state. One of these values: https://msdn.microsoft.com/library/windows/hardware/ff543130.aspx
- **InventoryVersion** List~~The version~~ of hardware ids for the device~~inventory file generating the events~~.
- **LowerClassFilters** Lower filter class drivers IDs installed for the device~~.~~
- **LowerFilters** Lower filter drivers IDs installed for the device
- **Manufacturer** INF file name (~~The device manufacturer MatchingID Represents~~ the name could be renamed by OS, such asoemX X.inf)
- **MatchingID** Device installation state.~~hardware ID or compatible ID that Windows uses to install a device instance~~
- **Model** The version~~device model ParentId Device instance id~~ of the inventory binary generating~~parent of~~ the events.

- **ParentId** Lower filter class drivers IDs installed for the device.
- **ProblemCode** Lower filter drivers IDs installed~~The current error code~~ for the device.
- **Provider** The device manufacturer.~~provider~~
- **Service** The device service name
- **STACKID** Represents~~A JSON array that provides~~ the hardware ID or compatible ID that Windows uses to install a~~value and order of the STACKID tree for the~~ device instance.~~See STACKID .~~
- **UpperClassFilters** Upper filter~~class~~ drivers IDs installed for the device
- **UpperFilters** The~~Upper filter drivers IDs installed for the~~ device model.

Microsoft.Windows.Inventory.Core.InventoryDevicePnpRemove

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

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.

This event includes fields from Ms.Device.DeviceInventoryChange.

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.InventoryDeviceUsbHubClassStartSync

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

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

Microsoft.Windows.Inventory.Core.InventoryDriverBinaryAdd

This event provides the basic metadata about driver binaries running on the system.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **DriverCheckSum** The checksum of the driver file.
- **DriverCompany** The company name that developed the driver.
- **DriverInBox** Is the driver included with the operating system?
- **DriverIsKernelMode** Is it a kernel mode driver?
- **DriverName** The file name of the driver.
- **DriverPackageStrongName** The strong name of the driver package
- **DriverSigned** The strong name of the driver package
- **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.
- **DriverVersion** The version of the driver file.
- **ImageSize** The size of the driver file.
- **Inf** The name of the INF file.
- **InventoryVersion** The version of the inventory file generating the events.
- **Product** The product name that is included in the driver file.
- **ProductVersion** The product version that is included in the driver file.
- **Service** The name of the service that is installed for the device.
- **WdfVersion** The Windows Driver Framework version.

Microsoft.Windows.Inventory.Core.InventoryDriverBinaryRemove

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

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

Microsoft.Windows.Inventory.Core.InventoryDriverPackageRemove

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

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

- **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.

This event includes fields from Ms.Device.DeviceInventoryChange.

The following fields are available:

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

Microsoft.Windows.Inventory.General~~Indicators~~.AppHealthStaticAdd~~Checksum~~

This
event sends details collected~~summarizes the counts~~ for a specific application~~the InventoryMiscellaneousUexIndicatorAdd events. The following fields are available: ChecksumDictionary A count of each operating system indicator. See ChecksumDictionary . PCFP Equivalent to the InventoryId field that is found in other core events. Microsoft.Windows.Inventory.Indicators.InventoryMiscellaneousUexIndicatorAdd These events represent the basic metadata about the OS indicators installed~~ on
the source~~system which are used for keeping the~~ device~~up to date~~.

~~This event includes fields from Ms.Device.DeviceInventoryChange .~~ The following fields are available:

- **AhaVersion~~IndicatorValue~~** ~~The~~ binary version~~indicator value. Microsoft.Windows.Inventory.Indicators.InventoryMiscellaneousUexIndicatorRemove This event is a counterpart to InventoryMiscellaneousUexIndicatorAdd that indicates that the item has been removed. This event includes fields from Ms.Device.DeviceInventoryChange . Microsoft.Windows.Inventory.Indicators.InventoryMiscellaneousUexIndicatorStartSync This event indicates that a new set of~~ of~~InventoryMiscellaneousUexIndicatorAdd events will be sent. This event includes fields from Ms.Device.DeviceInventoryChange . STACKID This event provides~~ the App Health Analyzer tool~~internal compatible ID for the stack~~.
- **ApplicationErrors** The count~~following fields are available: Order The index~~ of application errors from the event log~~ordered array~~.
- **BitnessValue** ~~The~~ architecture type~~value contained in the ordered array. Kernel events IO This event indicates the number~~of~~bytes read from or read by~~ the application (16 Bit~~OS and written to~~ or 32 bit~~written by the OS upon system startup. The following fields are available: BytesRead The total number of bytes read from~~ or 64 bit).
- **device_level** Various JRE/JAVA versions installed on a particular device~~read by the OS upon system startup~~.

- **ExtendedProperties** Attribute used for aggregating all other attributes under this event type.
- **Jar** FlagBytesWritten The total number of bytes written to determine if an app has a Java JAR file dependencyor written by the OS upon system startup.
- **Jre** FlagMicrosoft.Windows.Kernel.BootEnvironment.OsLaunch This event includes basic data about the Operating System, collected during Boot and used to determine if an app has JRE framework dependencyevaluate the success of the upgrade process.
- **Jre_version** JRE versions an app has declared framework dependency forThe following fields are available: BootApplicationId This field tells us what the OS Loader Application Identifier is.
- **Name** NameBootAttemptCount The number ofconsecutive times the applicationboot manager has attempted to boot into this operating system.
- **NonDPIAware** FlagBootSequence The current Boot ID, used to determine if an app is non-DPI awarecorrelate events related to a particular boot session.
- **NumBinaries** CountBootStatusPolicy Identifies the applicable Boot Status Policy. BootType Identifies the type of all binaries(.sys,.dll,.ini) from application install locationboot (e.
- **RequiresAdmin** Flagg.: "Cold", "Hiber", "Resume"). EventTimestamp Seconds elapsed since an arbitrary time point. This can be used to determine if an app requests admin privilegesidentify the time difference in successive boot attempts being made. FirmwareResetReasonEmbeddedController Reason for executionsystem reset provided by firmware.
- **RequiresAdminv2FirmwareResetReasonEmbeddedControllerAdditional** Additional flag to determineinformation on system reset reason provided by firmware if an app requests admin privilegesneeded. FirmwareResetReasonPch Reason for executionsystem reset provided by firmware.
- **RequiresUIAccessFirmwareResetReasonPchAdditional Additional information on system reset reason provided by firmware if needed. FirmwareResetReasonSupplied** Flag to determine if an app is based on UI featuresindicating that a reason for accessibilitysystem reset was provided by firmware.
- **VB6** FlagIO Amount of data written to determine if an app is based on VB6 frameworkand read from the disk by the OS Loader during boot.
- **VB6v2** Additional flagSee IO . LastBootSucceeded Flag indicating whether the last boot was successful. LastShutdownSucceeded Flag indicating whether the last shutdown was successful. MenuPolicy Type of advanced options menu that should be shown to determine if an appthe user (Legacy, Standard, etc.). RecoveryEnabled Indicates whether recovery is based on VB6 frameworkenabled.
- **Version** VersionUserInputTime The amount oftime theloader applicationspent waiting for user input.
- **VersionCheck** Flag to determine if an app has a static dependency on OS version.
- **VersionCheckv2** Additional flag to determine if an app has a static dependency on OS version.

OneDrive events Microsoft.WindowsOneDrive.InventorySync.GeneralSetup.AppHealthStaticStartSyncAPIOperation

This
event indicates the beginning of a series of AppHealthStaticAdd eventsincludes basic data about install and uninstall OneDrive API operations.

The following fields are available:

- **AllowTelemetry** Indicates the presenceAPIName The name of the 'allowtelemetry' command line argumentAPI.
- **CommandLineArgs** Command line arguments passed when launchingDuration How long the App Health Analyzerexecutableoperation took.
- **Enhanced** IndicatesIsSuccess Was the presenceoperation successful? ResultCode The result code. ScenarioName The name of the 'enhanced' command line argumentscenario.
- **StartTime** UTC date and time at which this event was sent.

Microsoft.WindowsOneDrive.InventorySync.GeneralSetup.InventoryMiscellaneousOfficeAddInAddEndExperience

Provides data onThis event includes a success or failure summary of the installed Office Add-insinstallation.

This event includesThe following fields fromare available: MsAPIName The name of the API.DeviceHResult Indicates the result code of the event IsSuccess Was the operation successful? ScenarioName The name of the scenario.DeviceInventoryChangeMicrosoft.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:

- **AddinCLSIDCurrentOneDriveVersion** The CLSID forcurrent version of OneDrive. CurrentOSBuildBranch The current branch ofth e Office add-inoperating system.
- **AddInId** Office add-in IDCurrentOSBuildNumber The current build number of the operating system.
- **AddinType** Office add-in TypeCurrentOSVersion The current version of the operating system.
- **BinFileTimestamp** TimestampHResult The HResult of the Office add-inoperation.
- **BinFileVersion** VersionSourceOSBuildBranch The source branch of the Office add-inoperating system.
- **Description** Office add-in description.
- **FileId** FileIdSourceOSBuildNumber The source build number of the Office add-inoperating system.
- **FileSize** File sizeSourceOSVersion The source version of the Office add-inoperating system.
- **FriendlyName** Friendly name for office add-inMicrosoft.
- **FullPath** Unexpanded pathOneDrive.Sync.Setup.RegisterStandaloneUpdaterAPIOperation This event is related toregistering or u nregistering the office add-inOneDrive update task.
- **InventoryVersion** The versionfollowing fields are available: APIName The name of the inventory binary generating theeventsAPI.
- **LoadBehavior** Uint32 that describesIsSuccess Was the load behavioroperation successful? RegisterNewTaskResult The HResult o f the RegisterNewTask operation.
- **OfficeApplicationScenarioName** The office application for this add-in.
- **OfficeArchitecture** Architecturename of the add-inscenario.
- **OfficeVersionUnregisterOldTaskResult** The office version for this add-inHResult of the UnregisterOldTask operation.
- **OutlookCrashingAddin** BooleanMicrosoft.OneDrive.Sync.Setup.SetupCommonData This event contains basic OneDrive configur ation data that indicates if crashes have been found for this add-inhelps to diagnose failures.
- **ProductCompany** The namefollowing fields are available: AppVersion The version of the company associated withapp. BuildArchitecture Is the Office add-in.
- **ProductNamearchitecture x86 or x64? Environment Is the device on the production or int service? MachineGuid** The prod uct name associated with the Office add-inCEIP machine ID.
- **ProductVersionMarket Which market is this in? MSFTInternal Is this an internal Microsoft device? OfficeVersionString** Th e version associated with theof Office add-inthat is installed.
- **ProgramId** The unique program identifier ofOSDeviceName Only if the Office add-indevice is internal to Microsoft, the device name.
- **Provider** Name ofOSUserName Only if the provider for this add-indevice is internal to Microsoft, the user name.UserGuid The CEIP user ID.

Microsoft.WindowsOneDrive.InventorySync.GeneralUpdater.InventoryMiscellaneousOfficeAddInRemoveCommon Data

IndicatesThis event contains basic OneDrive configuration data thathelps to diagnose failures. The following fields are available: AppVer sion The version of the app. BuildArch Is the architecture x86 or x64? Environment Is the device on the production or int service? IsMSFT Internal Is this particular data object represented byan internal Microsoft device? MachineGuid The CEIP machine ID. Market Which mar ket is this in? OfficeVersion The version of Office that is installed. OneDriveDeviceId The OneDrive device ID. OSDeviceName Only if the objectInstanceIddevice is no longer presentinternal to Microsoft, the device name.OSUserName Only if the device is internal to Microsof t, the user name. UserGuid A unique global user identifier. Microsoft.OneDrive.Sync.Updater.ComponentInstallState

This event
includes fields from Msbasic data about the installation state of dependent OneDrive components.Device.DeviceInventoryChange.

The following fields are available:

- **InventoryVersionComponentName** The versionname of
the inventory binary generatingdependent component. isInstalled Isthe events.dependent component installed?

Microsoft.WindowsOneDrive.InventorySync.GeneralUpdater.InventoryMiscellaneousOfficeAddInStartSyncOfficeRe gistration

This event indicates that a new sync is being generated for this object typethe status of the OneDrive integration with Microsoft Office.

This event includesThe following fields fromare available: MsisValid Is the Microsoft Office registration valid? Microsoft.DeviceOneDrive.DeviceInventoryChangeSync.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:

- **InventoryVersion32bit** The versionstatus of the inventory binary generatingOneDrive overlay icon on a 32-bit operating system. 64bit The status of the eventsOneDrive overlay icon on a 64-bit operating system.

Microsoft.WindowsOneDrive.InventorySync.GeneralUpdater.InventoryMiscellaneousOfficeIdentifiersAddRepairResult

Provides data onThe event determines the Office identifiersresult of the installation repair.

This event includesThe following fields fromare available: Mshr The HResult of the operation.DeviceMicrosoft.DeviceInventoryChangeOneDrive.Sync.Updater.SetupBinaryDownloadHResult This event indicates the status when downloading the OneDrive setup file.

The following fields are available:

- **InventoryVersionhr** The versionHResult of the inventory binary generating the eventsoperation.
- **OAudienceData** Sub-identifier for Microsoft Office release management,identifying.OneDrive.Sync.Updater.UpdateOverallResult This event determines the pilot group for a device
- **OAudienceId** Microsoft Office identifier for Microsoft Office release management, identifyingoutcome of the pilot group for a device
- **OMID** Identifier foroperation. The following fields are available: hr The HResult of the Office SQM Machine
- **OPlatform** Whether the installed Microsoft Office product is 32-bit or 64-bit
- **OTenantId** Unique GUID representing the Microsoft O365 Tenant
- **OVersion** Installedoperation. IsLoggingEnabled Is logging enabled? UpdaterVersion The version of Microsoft Officethe updater. For example, 16.0.8602.1000
- **OWowMID** Legacy Microsoft Office telemetry identifier (SQM Machine ID) for WoW systems (32-bit Microsoft Office on 64-bitWindows)

Microsoft.WindowsOneDrive.InventorySync.GeneralUpdater.InventoryMiscellaneousOfficeIdentifiersStartSyncUpdateTierReg

DiagnosticThis event to indicate a new sync is being generated for this object typedetermines status of the update tier registry values.

This event includesThe following fields fromare available: MsregReadEnterpriseHr The HResult of the enterprise reg read value.DeviceregReadTeamHr The HResult of the team reg read value.DeviceInventoryChangeMicrosoft.OneDrive.Sync.Updater.UpdateXmlDownloadHResult This event determines the status when downloading the OneDrive update configuration file.

The following fields are available:

- **InventoryVersionhr** The versionHResult of the inventory binary generating the eventsoperation.

Microsoft.WindowsOneDrive.InventorySync.GeneralUpdater.InventoryMiscellaneousOfficeIESettingsAddWebConnectionStatus

Provides data on Office-relatedThis event determines the error code that was returned when verifying Internet Explorerfeaturesconnectivity.

This event includesThe following fields fromare available: MswinInetError The HResult of the operation.DeviceRemediation events Microsoft.DeviceInventoryChangeWindows.Remediation.Applicable This event indicates a remedial plug-in is applicable if/when such a plug-in is detected.This is used to ensure Windows is up to date.

The following fields are available:

- **InventoryVersion** **ActionName** The version name of the inventory binary generating action to be taken by the events plug-in.
- **OIeFeatureAddon** Flag indicating which Microsoft Office products have this setting AppraiserBinariesValidResult Indicates whether plug-in was appraised as valid. AppraiserDetectCondition Indicates whether the plug-in passed the appraiser's check. AppraiserRegistryValidResult Indicates whether the registry entry checks out as valid. AppraiserTaskDisabled Indicates the appraiser task is disabled. AppraiserTaskValidFailed Indicates the Appraiser task did not function and requires intervention. CV Correlation vector DateTimeDifference The difference between local and reference clock times. DateTimeSyncEnabled Indicates whether the datetime sync plug-in is enabled. DaysSinceLastSIH The FEATURE_ADDON_MANAGEMENT feature letsapplications hosting number of days since the WebBrowser Control to respect add-on management selections madeusing most recent SIH executed. DaysToNextSIH The number of days until the Add-on Manager feature of InternetExplorer next scheduled SIH execution. Add-ons disabled by DetectedCondition Indicates whether detect condition is true and the user or by administrative group policy perform action will also be disabled in applications run. EvalAndReportAppraiserBinariesFailed Indicates the EvalAndReportAppraiserBinaries event failed. EvalAndReportAppraiserRegEntries Indicates the EvalAndReportAppraiserRegEntriesFailed event failed. EvalAndReportAppraiserRegEntriesFailed Indicates the EvalAndReportAppraiserRegEntriesFailed event failed. GlobalEventCounter Client side counter that enable this feature indicates ordering of events sent by the remediation system.
- **OIeMachineLockdown** Flag indicating which Microsoft Office products have this setting enabled HResult The HRESULT for detection or perform action phases of the plugin. When IsAppraiserLatestResult The HRESULT from the FEATURE_LOCALMACHINE_LOCKDOWN feature appraiser task. IsConfigurationCorrected Indicates whether the configuration of SIH task was successfully corrected. LastHresult The HRESULT for detection or perform action phases of the plugin. LastRun The date of the most recent SIH run. NextRun Date of the next scheduled SIH run. PackageVersion The version of the current remediation package. PluginName Name of the plugin specified for each generic plugin event. Reload True if SIH reload is required. RemediationNoisyHammerAcLineStatus Event that indicates the AC Line Status of the machine. RemediationNoisyHammerAutoStartCount The number of times hammer auto-started. RemediationNoisyHammerCalendarTaskEnabled Event that indicates Update Assistant Calendar Task is enabled, Internet Explorer applies security restrictions on content loaded from. RemediationNoisyHammerCalendarTaskExists Event that indicates an Update Assistant Calendar Task exists. RemediationNoisyHammerCalendarTaskTriggerEnabledCount Event that indicates calendar triggers are enabled in the user's local task. RemediationNoisyHammerDaysSinceLastTaskRunTime The number of days since the most recent hammer task ran. RemediationNoisyHammerGetCurrentSize Size in MB of the $GetCurrent folder. RemediationNoisyHammerIsInstalled TRUE if the noisy hammer is installed. RemediationNoisyHammerLastTaskRunResult The result of the last hammer task run. RemediationNoisyHammerMeteredNetwork TRUE if the machine, which helps prevent malicious behavior involving local files is on a metered network.
- **OIeMimeHandling** Flag indicating which Microsoft Office products have this setting RemediationNoisyHammerTaskEnabled Indicates whether the Update Assistant Task (Noisy Hammer) is enabled. When RemediationNoisyHammerTaskExists Indicates whether the FEATURE_MIME_HANDLING feature control Update Assistant Task (Noisy Hammer) exists. RemediationNoisyHammerTaskTriggerEnabledCount Indicates whether counting is enabled, Internet Explorer handles MIMEtypes more securely for the Update Assistant (Noisy Hammer) task trigger. Only applies to Windows Internet Explorer6 RemediationNoisyHammerUAExitCode The exit code of the Update Assistant (Noisy Hammer) task. RemediationNoisyHammerUAExitState The code for Windows XP Service Pack 2 the exit state of the Update Assistant (SP2 Noisy Hammer)
- **OIeMimeSniffing** Flag indicating which Microsoft Office products have this setting enabled task. Determines RemediationNoisyHammerUserLoggedIn TRUE if there is a file's type by examining its bit signature user logged in. RemediationNoisyHammerUserLoggedInAdmin TRUE if there is the user currently logged in is an Admin. RemediationShellDeviceManaged TRUE if the device is WSUS managed or Windows Internet Explorer uses this information todetermine how to render Updated disabled. RemediationShellDeviceNewOS TRUE if the file device has a recently installed OS. The FEATURE RemediationShellDeviceSccm TRUE if the device is managed by SCCM (Microsoft System Center Configuration Manager). RemediationShellDeviceZeroExhaust TRUE if the device has opted out of Windows Updates completely. RemediationTargetMachine Indicates whether the device is a target of the specified fix. RemediationTaskHealthAutochkProxy True/False based on the health of the AutochkProxy task. RemediationTaskHealthChkdskProactiveScan True/False based on the health of the Check Disk task. RemediationTaskHealthDiskCleanup_MIMESilentCleanup True/False based on the health of the Disk Cleanup task. RemediationTaskHealthMaintenance_SNIFFING feature, when WinSAT True/False based on the health of the Health Maintenance task. RemediationTaskHealthServicing_ComponentCleanupTask True/False based on the health of the Health Servicing Component task. RemediationTaskHealthUSO_ScheduleScanTask True/False based on the health of the USO (Update Session Orchestrator) Schedule task. RemediationTaskHealthWindowsUpdate_ScheduledStartTask True/False based on the health of the Windows Update Scheduled Start task. RemediationTaskHealthWindowsUpdate_SihbootTask True/False based on the health of the Sihboot task. RemediationUHServiceBitsServiceEnabled Indicates whether BITS service is enabled, allows. RemediationUHServiceDeviceInstallEnabled Indicates whether Device Install service is enabled. Remediat

ionUHServiceDoSvcServiceEnabled Indicates whether DO service is enabled. RemediationUHServiceDsmsvcEnabled Indicates whether DSMSVC service is enabled. RemediationUHServiceLicensemanagerEnabled Indicates whether License Manager service is enabled. RemediationUHServiceMpssvcEnabled Indicates whether MPSSVC service is enabled. RemediationUHServiceTokenBrokerEnabled Indicates whether Token Broker service is enabled. RemediationUHServiceTrustedInstallerServiceEnabled Indicates whether Trusted Installer service is enabled. RemediationUHServiceUsoServiceEnabled Indicates whether USO (Update Session Orchestrator) service is enabled. RemediationUHServicew32timeServiceEnabled Indicates whether W32 Time service is enabled. RemediationUHServiceWecsvcEnabled Indicates whether WECSVC service is enabled. RemediationUHServiceWinmgmtEnabled Indicates whether WMI service is enabled. RemediationUHServiceWpnServiceEnabled Indicates whether WPN service is enabled. RemediationUHServiceWuauservServiceEnabled Indicates whether WUAUSERV service is enabled. Result This is the HRESULT for Detection or Perform Action phases of the plugin. RunAppraiserFailed Indicates RunAppraiser failed torun correctly. RunTask TRUE if SIH task should be set differentlyrun by the plug-in. TimeServiceNTPServer The URL for eachsecurity zonethe NTP time server used by usingdevice. TimeServiceStartType The start up type for the URLACTION_FEATURE_MIME_SNIFFING URL action flag

- **OIeNoAxInstall** Flag indicating whichNTP time service. TimeServiceSyncDomainJoined True if device domain joined and hence uses DC for clock. TimeServiceSyncType Type of sync behavior for Date & Time service on device. Microsoft Office productshave this setting enabled. WhenWindows.Remediation.Completed This event enables completion tracking of a webpageattempts to load or install an ActiveX controlprocess that isn't already installed,remediates issues preventing security and quality updates. The following fields are available: ActionName Name of the FEATURE_RESTRICT_ACTIVEXINSTALL featureblocksaction to be completed by the requestplug-in. When a webpage triesAppraiserTaskCreationFailed TRUE if the appraiser task creation failed to load or install an ActiveX control that isn't already installed,complete successfully. AppraiserTaskDeleteFailed TRUE if deletion of appraiser task failed to complete successfully. AppraiserTaskExistFailed TRUE if detection of the FEATURE_RESTRICT_ACTIVEXINSTALLappraiser task failed to complete successfully. AppraiserTaskLoadXmlFailed TRUE if the Appraiser XML Loader failed to complete successfully. AppraiserTaskMissing TRUE if the Appraiser task is missing. AppraiserTaskTimeTriggerUpdateFailedId TRUE if the Appraiser Task Time Trigger failed to update successfully. AppraiserTaskValidateTaskXmlFailed TRUE if the Appraiser Task XML failed to complete successfully. CrossedDiskSpaceThreshold Indicates if cleanup resulted in hard drive usage threshold required for feature blocksupdate to be exceeded. CV The Correlation Vector. DateTimeDifference The difference between the request
- **OIeNoDownload** Flag indicating which Microsoft Office products have this setting enabledlocal and reference clocks.DaysSinceOsInstallation The FEATURE_RESTRICT_FILEDOWNLOAD feature blocks file download requests that navigateto a resourcenumber of days since the installation of the Operating System. DiskMbCleaned The amount of space cleaned on the hard disk, that display a file download dialog boxmeasured in Megabytes. DiskMbFreeAfterCleanup The amount of free hard disk space after cleanup, ormeasured in Megabytes. DiskMbFreeBeforeCleanup The amount of free hard disk space before cleanup, measured in Megabytes. ForcedAppraiserTaskTriggered TRUE if Appraiser task ran from the plug-in. GlobalEventCounter Client-side counter that are not initiated explicitlyindicates ordering of events sent by athe active user action (for example. HandlerCleanupFreeDiskInMegabytes The amount of hard disk space cleaned by the storage sense handlers, a mouse click or key press). Only applies to Windows Internet Explorer 6measured in Megabytes. HResult The result of the event execution. LatestState The final state of the plug-in component. PackageVersion The package version forthe current Remediation. PageFileCount The number of Windows XP Service Pack 2Page files. PageFileCurrentSize The size of the Windows Page file, measured in Megabytes. PageFileLocation The storage location (SP2directory path)
- **OIeObjectCaching** Flag indicating which Microsoft Office products have this setting enabledof the Windows Page file. WhenenabledPageFilePeakSize The maximum amount of hard disk space used by the Windows Page file,measured in Megabytes. PluginName The name of the FEATURE_OBJECT_CACHING feature prevents webpages from accessing or instantiating ActiveXcontrols cached from different domains or security contexts
- **OIePasswordDisable** Flag indicating which Microsoft Office products have this setting enabledplug-in specified for each generic plug-in event. AfterRanCleanup TRUE if the plug-in ran disk cleanup. RemediationConfigurationTroubleshooterExecuted True/False based on whether the Remediation Configuration Troubleshooter executed successfully. RemediationConfigurationTroubleshooterIpconfigFix TRUE if IPConfig Fix completed successfully. RemediationConfigurationTroubleshooterNetShFix TRUE if network card cache reset ran successfully. RemediationDiskCleanSizeBtWindowsFolderInMegabytes The size of the Windows Internet Explorer 6 forBT folder (used to store Windows XP Service Pack 2upgrade files), measured in Megabytes. RemediationDiskCleanupBTFolderEsdSizeInMB The size of the Windows BT folder (SP2used to store Windows upgrade files) ESD (Electronic Software Delivery), Internet Explorerno longer allows usernames and passwords to be specifiedmeasured in URLs that use the HTTP or HTTPSprotocolsMegabytes. URLs using other protocolsRemediationDiskCleanupGetCurrentEsdSizeInMB The size of any existing ESD (Electronic Software Delivery) folder, such as FTPmeasured in Megabytes. RemediationDiskCleanupSearchFileSizeInMegabytes The size of the Cleanup Search index file, still allow usernames and passwords
- **OIeSafeBind** Flag indicating which Microsoft Office products have this setting enabledmeasured in Megabytes.RemediationDiskCleanupUpdateAssistantSizeInMB The FEATURE_SAFE_BINDTOOBJECT feature performs additionalsafety checks when calling Mo

nikerBindToObject to create and initialize Microsoft ActiveX controlssize of the Update Assistant folder, measured in Megabytes. Specifically, prevent the control from being createdRemediationDoorstopChangeSucceeded TRUE if COMPAT_EVIL_DONT_LOAD Doorstop registry key was successfully modified. RemediationDoorstopExists TRUE if thereis ina OneSettings Doorstop value. RemediationDoorstopRegkeyError TRUE if an error occurred accessing theDoorstopregistry forkey. RemediationDRFKeyDeleteSucceeded TRUE if the control

- **OIeSecurityBand** Flag indicating which Microsoft Office products have this setting enabledRecoveredFrom (Doorstop) registry key was successfully deleted.RemediationDUABuildNumber The FEATURE_SECURITYBAND feature controlsbuild number of the displayDUA. RemediationDUAKeyDeleteSucceeded TRUE if the UninstallActive registry key was successfully deleted. RemediationDuplicateTokenSucceeded TRUE if the user token was successfully duplicated. RemediationImpersonateUserSucceeded TRUE if the user was successfully impersonated. RemediationNoisyHammerTaskKickOffIsSuccess TRUE if the NoisyHammer task started successfully. RemediationQueryTokenSucceeded TRUE if the user token was successfully queried. RemediationRanHibernation TRUE if the system entered Hibernation. RemediationRevertToSystemSucceeded TRUE if reversion to the system context succeeded. RemediationUpdateServiceHealthRemediationResult The result of the Internet Explorer Information barUpdate Service Health plug-in. When enabledRemediationUpdateTaskHealthRemediationResult The result of the Update Task Health plug-in. RemediationUpdateTaskHealthTaskList A list of tasks fixed by the Update Task Health plug-in. RemediationWindowsLogSpaceFound The size of the Windows log files found,measured in Megabytes. RemediationWindowsLogSpaceFreed The amount of disk space freed by deleting the Information bar appears when filedownloadWindows log files, measured in Megabytes. RemediationWindowsSecondaryDriveFreeSpace The amount of free space on the secondary drive, measured in Megabytes. RemediationWindowsSecondaryDriveLetter The letter designation of the first secondary drive with a total capacity of 10GB or code installation is restricted
- **OIeUncSaveCheck** Flag indicating which Microsoft Office products have this settingenabledmore.RemediationWindowsSecondaryDriveTotalSpace The FEATURE_UNC_SAVEDFILECHECK feature enablestotal storage capacity of the Marksecondary drive, measured in Megabytes. RemediationWindowsTotalSystemDiskSize The total storage capacity of the Web (MOTW)System Disk Drive, measured in Megabytes. Result The HRESULT for local files loadedfrom network locations that have been shared by usingDetection or Perform Action phases of the Universal NamingConvention (UNC)
- **OIeValidateUrl** Flag indicating which Microsoft Office products have this setting enabledplug-in. When enabledRunResult The HRESULT for Detection or Perform Action phases of the plug-in. ServiceHealthPlugin The nae of the Service Health plug-in. StartComponentCleanupTask TRUE if the Component Cleanup task started successfully. TotalSizeofOrphanedInstallerFilesInMegabytes The size of any orphaned Windows Installer files,measured in Megabytes. TotalSizeofStoreCacheAfterCleanupInMegabytes The size of the FEATURE_VALIDATE_NAVIGATE_URL feature control preventsWindows Internet Explorer from navigatingStore cache after cleanup, measured in Megabytes. TotalSizeofStoreCacheBeforeCleanupInMegabytes The size of the Windows Store cache (prior to a badly formed URL
- **OIeWebOcPopup** Flag indicating which Microsoft Office products have this setting enabledcleanup), measured in Megabytes.usoScanDaysSinceLastScan The FEATURE_WEBOC_POPUPMANAGEMENT feature allows applicationshostingnumber of days since the WebBrowser Controllast USO (Update Session Orchestrator) scan. usoScanInProgress TRUE if a USO (Update Session Orchestrator) scan is in progress, to receiveprevent multiple simultaneous scans. usoScanIsAllowAutoUpdateKeyPresent TRUE if the default Internet Explorer pop-up window management behavior
- **OIeWinRestrict** Flag indicating which Microsoft Office products have this setting enabledAllowAutoUpdate registry key is set. When enabledusoScanIsAllowAutoUpdateProviderSetKeyPresent TRUE if AllowAutoUpdateProviderSet registry key is set. usoScanIsAuOptionsPresent TRUE if Auto Update Options registry key is set. usoScanIsFeatureUpdateInProgress TRUE if a USO (Update Session Orchestrator) scan is in progress,to prevent multiple simultaneous scans. usoScanIsNetworkMetered TRUE if the FEATURE_WINDOW_RESTRICTIONS feature adds several restrictionsdevice is currently connected toa metered network. usoScanIsNoAutoUpdateKeyPresent TRUE if no Auto Update registry key is set/present. usoScanIsUserLoggedOn TRUE if theuser is logged on. usoScanPastThreshold TRUE if the most recent USO (Update Session Orchestrator) scan is past the threshold (late). usoScanType The type of USO (Update Session Orchestrator) scan (Interactive or Background). WindowsHyberFilSysSizeInMegabytes The size and behavior of popup windows
- **OIeZoneElevate** Flag indicating which Microsoft Office products have this setting enabledthe Windows Hibernation file, measured in Megabytes. When enabledWindowsInstallerFolderSizeInMegabytes The size of the Windows Installer folder,measured in Megabytes. WindowsOldFolderSizeInMegabytes The size of the FEATURE_ZONE_ELEVATION featureprevents pagesWindows.OLD folder, measured in one zone from navigating to pagesMegabytes. WindowsOldSpaceCleanedInMB The amount of disk space freed by removing the Windows.OLD folder, measured in a highersecurity zone unlessMegabytes. WindowsPageFileSysSizeInMegabytes The size of the navigation is generated byWindows Page file, measured in Megabytes. WindowsSoftwareDistributionFolderSizeInMegabytes The size of the userSoftwareDistribution folder, measured in Megabytes. WindowsSwapFileSysSizeInMegabytes The size of the Windows Swap file, measured in Megabytes. WindowsSxsFolderSizeInMegabytes The size of the WinSxS (Windows Side-by-Side) folder, measured in Megabytes. WindowsSxsTempFolderSizeInMegabytes The size of the WinSxS (Windows Side-by-Side) Temp folder, measured in Megabytes.

Microsoft.Windows.InventoryRemediation.General.InventoryMiscellaneousOfficeIESettingsStartSyncDiskCleanUnExpectedErrorEvent

DiagnosticThis eventindicates that an unexpected error occurred during an update and provides information to indicate a new syncis being generated for this object typehelp address the issue.

This event includesThe following fields fromare available: MsCV The Correlation vector.DeviceErrorMessage A description of any errors encountered while the plug-in was running.DeviceInventoryChangeGlobalEventCounter The client-side counter that indicates ordering of events. Hresult The result of the event execution. PackageVersion The version number of the current remediation package. SessionGuid GUID associated with a given execution of sediment pack. Microsoft.Windows.Remediation.Error This event indicates a Sediment Pack error (update stack failure) has been detected and provides information to help address the issue.

The following fields are available:

- **InventoryVersionHResult** The versionresult of the inventory binary generatingevent execution. Message A message containing information about the eventserror that occurred .PackageVersion The version number of the current remediation package.

Microsoft.Windows.InventoryRemediation.General.InventoryMiscellaneousOfficeInsightsAddFallbackError

This eventindicates an error when Self Update results in a Fallback and provides insight data oninformation to help address the installed Office productsissue.

This event includesThe following fields fromare available: Mss0 Indicates the Fallback error level.DeviceSee Microsoft.DeviceInventoryChangeWindows.Remediation.wilResult .wilResult The result of the Windows Installer Logging. See wilResult

. Microsoft.Windows.Remediation.RemediationNotifyUserFixIssuesInvokeUIEvent This event occurs when the Notify User task executes and provides information about the cause of the notification. The following fields are available:

- **InventoryVersionCV** The versionCorrelation vector. GlobalEventCounter The client-side counter that indicates ordering of theinventory binary generating the events.
- **OfficeApplicationPackageVersion** The nameversion number of the Office applicationcurrent remediation package.
- **OfficeArchitectureRemediationNotifyUserFixIssuesCallResult** The bitnessresult ofcalling the Office applicationUSO (Update Session Orchestrator) sequence steps.
- **OfficeVersionRemediationNotifyUserFixIssuesUsoDownloadCalledHr** The version oferror code from the OfficeapplicationUSO (Update Session Orchestrator) download call.
- **ValueRemediationNotifyUserFixIssuesUsoInitializedHr** The insights collected about this entityerror code from the USO (Update Session Orchestrator) initialize call.RemediationNotifyUserFixIssuesUsoProxyBlanketHr The error code from the USO (Update Session Orchestrator) proxy blanket call. RemediationNotifyUserFixIssuesUsoSetSessionHr The error code from the USO (Update Session Orchestrator) session call.

Microsoft.Windows.InventoryRemediation.General.InventoryMiscellaneousOfficeInsightsRemoveRemediationShellFailedAutomaticAppUpdateModifyEventId

Indicates that this particular data object represented byThis event provides the objectInstanceId is no longer presentmodification of the date on which an Automatic App Update scheduled task failed and provides information about the failure.

This event includesThe following fields fromare available: MsCV The Correlation Vector.DeviceGlobalEventCounter The client-side counter that indicates ordering of events.DeviceInventoryChangehResult The result of the event execution. PackageVersion The version number of the current remediation package. Microsoft.Windows.Remediation.RemediationShellUnexpectedExceptionId This event identifies the remediation plug-in that returned an unexpected exception and provides information about the exception.

The following fields are available:

- **InventoryVersion~~CV~~** The~~Correlation Vector. GlobalEventCounter The client-side counter that indicates ordering of events. PackageVersion The~~ version~~number~~ of the inventory binary generating~~current remediation package. RemediationShellUnexpectedExceptionId The ID of~~ the events~~remediation plug-in that caused the exception~~.

Microsoft.Windows.Inventory~~Remediation~~.General.InventoryMiscellaneousOfficeInsightsStartSync~~RemediationUHEnableServiceFailed~~

This diagnostic event indicates that a new sync is being generated for this object type~~tracks the health of key update (Remediation) services and whether they are enabled~~.

This event includes~~The following~~ fields from~~are available:~~ Ms~~CV The Correlation Vector~~.Device~~GlobalEventCounter The client-side counter that indicates ordering of events~~.DeviceInventoryChange~~hResult The result of the event execution. PackageVersion The version number of the current remediation package. serviceName The name associated with the operation. Microsoft.Windows.Remediation.RemediationUpgradeSucceededDataEventId~~ This event returns information about the upgrade upon success to help ensure Windows is up to date~~.

The following fields are available:

- **InventoryVersion~~AppraiserPlugin TRUE / FALSE depending on whether the Appraiser plug-in task fix was successful. ClearAUOptionsPlugin TRUE / FALSE depending on whether the AU (Auto Updater) Options registry keys were successfully deleted. CV~~** The~~Correlation Vector. DatetimeSyncPlugin TRUE / FALSE depending on whether the DateTimeSync plug-in ran successfully. DiskCleanupPlugin TRUE / FALSE depending on whether the DiskCleanup plug-in ran successfully. GlobalEventCounter The client-side counter that indicates ordering of events. NoisyHammerPlugin TRUE / FALSE depending on whether the NoisyHammer plug-in ran successfully. PackageVersion The~~ version~~number~~ of the inventory binarygenerating~~current remediation package. RebootRequiredPlugin TRUE / FALSE depending on whether~~ the events~~Reboot plug-in ran successfully.RemediationNotifyUserFixIssuesPlugin TRUE / FALSE depending on whether the User Fix Issues plug-in ran successfully RemediationPostUpgradeDiskSpace The amount of disk space available after the upgrade. RemediationPostUpgradeHibernationSize The size of the Hibernation file after the upgrade. ServiceHealthPlugin A list of services updated by the plug-in. SIHHealthPlugin TRUE / FALSE depending on whether the SIH Health plug-in ran successfully. StackDataResetPlugin TRUE / FALSE depending on whether the update stack completed successfully. TaskHealthPlugin A list of tasks updated by the plug-in. UpdateApplicabilityFixerPlugin TRUE / FALSE depending on whether the update applicability fixer plug-in completed successfully. WindowsUpdateEndpointPlugin TRUE / FALSE depending on whether the Windows Update Endpoint was successful.~~

Microsoft.Windows.Inventory~~Remediation~~.General.InventoryMiscellaneousOfficeProductsAdd~~Started~~

Describes Office Products installed~~This event reports whether a plug-in started, to help ensure Windows is up to date~~.

This event includes~~The following~~ fields from~~are available:~~ Ms~~CV The Correlation Vector~~.Device~~GlobalEventCounter The client-side counter that indicates ordering of events~~.DeviceInventoryChange~~PackageVersion The version number of the current remediation package. PluginName The name of the plug-in specified for each generic plug-in event. Result The HRESULT for Detection or Perform Action phases of the plug-in. Microsoft.Windows.Remediation.wilResult~~ This event provides Self Update information to help keep Windows up to date~~.

The following fields are available:

- **InventoryVersion** The version~~callContext A list of~~diagnostic activities containing this error. currentContextId An identifier for~~the inventory~~newest diagnostic activity containing this error. currentContextMessage A message associated with the most recent diagnostic activity containing this error (if any). currentContextName Name of the most recent diagnostic activity containing this error. failureCount Number of failures seen within the~~ binary generating~~where~~ the events~~error occurred~~.
- **OC2rApps** A GUID~~failureId The identifier assigned to this failure. failureType Indicates~~ the describes~~type of failure observed (exception, returned, error, logged error, or fail fast). fileName The source code file name where~~ the Office Click-To-Run apps

- **OC2rSkus** Comma-delimited list (CSV)~~error occurred. function The name~~ of Office Click-To-Run products installed on the device~~function where the error occurred~~. For example, Office 2016 ProPlus
- **OMsiApps** Comma-delimited list~~hresult The failure error code. lineNumber The Line Number within the source code file where the error occurred. message A message associated with the failure~~ (CSV)~~if any). module The name~~ of Office MSI productsinstalled on the device~~binary module in which the error occurred~~. For example, Microsoft Word
- **OProductCodes~~originatingContextId The identifier for the oldest diagnostic activity containing this error. originatingContextMessage~~** A GUID that describes~~message associated with~~ the Office MSI products~~oldest diagnostic activity containing this error (if any). originatingContextName The name of the oldest diagnostic activity containing this error. threadId The identifier of the thread the error occurred on.~~

~~Sediment events~~ Microsoft.Windows.Inventory~~Sediment~~.General~~Info~~.InventoryMiscellaneousOfficeProductsStartSyncAppraiserData~~

Diagnostic~~This~~ event~~provides data on the current Appraiser status of the device~~ to indicate a new sync~~help ensure Windows~~ is being generated~~up to date. The following fields are available: ErrorCode The value of the Return Code~~ for~~the registry query. GStatus The pre-upgrade GStatus value. PayloadVersion The version information for the remediation component. RegKeyName The name of the registry subkey where data was found for~~ this object type~~event.Time The system time at which the event began. UpgEx The pre-upgrade UpgEx value. Microsoft.Windows.Sediment.Info.BinaryInfo~~

This event includes~~provides information about the binary returned by the Operating System Remediation System Service (OSRSS) to help ensure Windows is up to date. The following fields~~ are available: BinaryPath The sanitized name of the system binary from Ms~~which the data was gathered~~.Device~~ErrorCode The value of the return code for querying the version from the binary~~.DeviceInventoryChange~~FileVerBuild The binary's build number. FileVerMajor The binary's major version number. FileVerMinor The binary's minor version number. FileVerRev The binary's revision number. PayloadVersion The version information for the remediation component. Time The system time at which the event began. Microsoft.Windows.Sediment.Info.DownloadServiceError~~ ~~This event provides information when the Download Service returns an error.The information provided helps keep Windows up to date.~~

The following fields are available:

- **InventoryVersion~~Architecture~~** The version~~platform architecture used to identify the correct download payload. BuildNumber The starting build number used to identify the correct download payload. Edition The Operating System Edition used to identify the correct download payload. Error The description~~ of the inventory binary generating~~error encountered. LanguageCode The system User Interface Language used to identify~~ the event s~~correct download payload.Stack Details about the error encountered. WorkingDirectory The folder location (path) downloader was attempting to say the payload to.~~

Microsoft.Windows.Inventory~~Sediment~~.General~~Info~~.InventoryMiscellaneousOfficeSettingsAdd~~DownloadServiceProgress~~

This event describes various Office settings~~indicates the progress of the downloader in 1% increments.~~

This event includes~~The following~~ fields from~~are available:~~ Ms~~Percentage The amount successfully downloaded, measured as a percentage of the whole~~.Device~~Microsoft~~.DeviceInventoryChange~~Windows.Sediment.Info.Error~~ ~~This event indicates an error in the updater payload.This information assists in keeping Windows up to date.~~

The following fields are available:

- **BrowserFlags** Browser flags for Office-related products~~FailureType The type of error encountered~~.
- **ExchangeProviderFlags** Provider policies for Office Exchange~~FileName The code file in which the error occurred~~.
- **InventoryVersion~~HResult~~** The~~failure error code. LineNumber The line number in the code file at which the error occurred. ReleaseVer The~~ version of~~information for~~ the inventory binary generating~~component in which~~ the events~~error occurred~~.
- **SharedComputerLicensing** Office shared computer licensing policies~~Time The system time at which the error occurred~~.

Microsoft.Windows.Inventory~~Sediment~~.General~~Info~~.InventoryMiscellaneousOfficeSettingsStartSync~~PhaseChange~~

Indicates a new sync is being generated for this object type~~The event indicates progress made by the updater.This information assists in keeping Windows up to date.~~

This event includes~~The following~~ fields from~~are available:~~ Ms~~NewPhase The phase of progress made.~~Device~~ReleaseVer The version information for the component in which the change occurred.~~DeviceInventoryChange~~Time The system time at which the phase chance occurred. Microsoft.Windows.Sediment.Info.ServiceInfo This event provide information about the system service for which data is being gathered by the Operating System Remediation System Service (OSRSS) to help ensure Windows is up to date~~.

The following fields are available:

- **InventoryVersion~~ErrorCode~~** The~~value returned by the error for querying the service information. PayloadVersion The~~version~~information for the remediation component. ServiceName The name~~ of the inventory binary generating~~system service for which data was gathered. ServiceStatus The status of~~ the events~~specified service.Time The system time at which the event occurred.~~

Microsoft.Windows.Inventory~~Sediment~~.General~~Info~~.InventoryMiscellaneousOfficeVBAAdd~~Uptime~~

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

~~information about how long the device has been operating.~~ This event includes fields from Ms~~information helps ensure Windows is up to date~~.Device.DeviceInventoryChange.

The following fields are available:

- **Design** Count~~Days The number~~ of files with design issues found~~days the device has been on~~.
- **Design_x64** Count~~Hours The number~~ of files with 64 bit design issues found~~hours the device has been on~~.
- **DuplicateVBA** Count~~Minutes The number~~ of files with duplicate VBA code~~minutes the device has been on~~.
- **HasVBA** Count of files with VBA code~~PayloadVersion The version information for the remediation component~~.
- **Inaccessible** Count~~Seconds The number~~ of files that were inaccessible for scanning~~seconds the machine has been on~~.
- **InventoryVersion~~Ticks~~** The version~~number of~~~~system clock ticks"~~ the inventory binary generating~~device has been on. Time The system time at which~~ the events~~event occurred~~.
- **Issues** Count of files with issues detected~~Microsoft~~.
- **Issues_x64** Count of files with 64-bit issues detected~~Windows~~.
- **IssuesNone** Count of files with no issues detected~~Sediment~~.
- **IssuesNone_x64** Count of files with no 64-bit issues detected~~OSRSS~~.
- **Locked** Count of files~~CheckingOneSettings This event indicates the parameters~~ that were locked, preventing scanning~~the Operating System Remediation System Service (OSRSS) uses for a secure ping to Microsoft to help ensure Windows is up to date~~.
- **NoVBA** Count of files with no VBA inside~~The following fields are available: CustomVer The registry value for targeting~~.
- **Protected** Count of files that were password protected, preventing scanning~~IsMetered TRUE if the machine is on a metered network~~.
- **RemLimited** Count~~LastVer The version~~ of files that require limited remediation changes~~the last successful run~~.
- **RemLimited_x64** Count~~ServiceVersionMajor The Major version information~~ of files that require limited remediation changes for 64-bit issues~~the component~~.
- **RemSignificant** Count~~ServiceVersionMinor The Minor version information~~ of files that require significant remediationchanges~~the component~~.
- **RemSignificant_x64** Count of files that require significant remediation changes for 64-bit issues~~Time The system time at which the event occurred~~.
- **Score** Overall compatibility score calculated for scanned content~~Microsoft~~.
- **Score_x64** Overall 64-bit compatibility score calculated for scanned content~~Windows~~.
- **Total** Total number of files scanned~~Sediment~~.
- **Validation** Count~~OSRSS.DownloadingUrl This event provides information about the URL from which the Operating System Remediation System Service (OSRSS) is attempting to download. This information helps ensure Windows is up to date. The following fields are available: AttemptNumber The count indicating which download attempt is starting. ServiceVersionMajor The Major version information~~ of files that require additional manual validation~~the component~~.

- **Validation_x64** CountServiceVersionMinor The Minor version information of files that require additional manual validation for64-bit issuesthe component.Time The system time at which the event occurred. Url The URL from which data was downloaded.

Microsoft.Windows.InventorySediment.GeneralOSRSS.InventoryMiscellaneousOfficeVBARemoveDownloadSuccess

Indicates that this particularThis event indicates the Operating System Remediation System Service (OSRSS) successfully downloaddata object represented byfrom the objectInstanceIdindicated URL. This information helps ensure Windows is no longer presentup to date.

This event includesThe following fields fromare available: MsServiceVersionMajor The Major version information of the component.DeviceServiceVersionMinor The Minor version information of the component.DeviceInventoryChangeTime The system time at which the event occurred. Url The URL from which data was downloaded. Microsoft.Windows.Sediment.OSRSS.Error This event indicates an error occurred in the Operating System Remediation System Service (OSRSS). The information provided helps ensure future upgrade/update attempts are more successful.

The following fields are available:

- **InventoryVersionFailureType** Thetype of error encountered. FileName The code file in which the error occurred. HResult The failure error code. LineNumber The line number in the code file at which the error occurred. ServiceVersionMajor The Majorversioninformation of
  the inventory binary generatingcomponent. ServiceVersionMinor The Minor version information of the eventscomponent.Time The system time at which the event occurred.

Microsoft.Windows.InventorySediment.GeneralOSRSS.InventoryMiscellaneousOfficeVBARuleViolationsAddExeSignatureValidated

This
event provides data on Microsoft Office VBA rule violations, including a rollup count per violation type, giving anindicationindicates the Operating System Remediation System Service (OSRSS) successfully validated the signature of remediationrequirements for an organizationEXE from the indicated URL.
The event identifierinformation provided helps ensure Windows is aunique GUID, associated with the validation ruleup to date.

This event includesThe following fields fromare available: MsServiceVersionMajor The Major version information of the component.DeviceServiceVersionMinor The Minor version information of the component.DeviceInventoryChangeTime The system time at which the event occurred. Url The URL from which the validated EXE was downloaded. Microsoft.Windows.Sediment.OSRSS.ExtractSuccess This event indicates that the Operating System Remediation System Service (OSRSS) successfully extracted downloaded content.The information provided helps ensure Windows is up to date.

The following fields are available:

- **Count** CountServiceVersionMajor The Major version information of total Microsoft Office VBA rule violations
- **InventoryVersionthe component. ServiceVersionMinor** TheMinor versioninformation of
  the inventory binarygeneratingcomponent. Time The system time at which the eventsevent occurred.Url The URL from which the successfully extracted content was downloaded.

Microsoft.Windows.InventorySediment.GeneralOSRSS.InventoryMiscellaneousOfficeVBARuleViolationsRemoveNewUrlFound

Indicates that this particular data object represented byThis event indicates the objectInstanceIdOperating System Remediation System Service (OSRSS) succeeded in finding a new URL to download from. This helps ensure Windows is no longer presentup to date.

This event includesThe following fields fromare available: MsServiceVersionMajor The Major version information of the component.DeviceServiceVersionMinor The Minor version information of the component.DeviceInventoryChangeTime The system time at which the event occurred. Url The new URL from which content will be downloaded. Microsoft.Windows.Sediment.OSRSS.ProcessCreated This event in

dicates the Operating System Remediation System Service (OSRSS) created a new process to execute content downloaded from the indicated URL.This information helps ensure Windows is up to date.

The following fields are available:

- **InventoryVersionServiceVersionMajor** TheMajor versioninformation of the inventory binary generatingcomponent. ServiceVersionMinor The Minor version information of the eventscomponent.Time The system time at which the event occurred. Url The new URL from which content will be executed.

Microsoft.Windows.InventorySediment.GeneralOSRSS.InventoryMiscellaneousOfficeVBARuleViolationsStartSyncUrlState

This event indicates thatthe state the Operating System Remediation System Service (OSRSS) is in while attempting a new sync isbeing generated for this object typedownload from the URL.

This event includesThe following fields fromare available: MsId A number identifying the URL ServiceVersionMajor Version info for the component ServiceVersionMinor Version info for the component StateData State-specific data, such as which attempt number for the download StateNumber A number identifying which state the URL is in (found, downloading, extracted, etc.) Time System timestamp the event was fired Microsoft.DeviceWindows.DeviceInventoryChangeSediment.ServiceInstaller.AttemptingUpdate This event indicates the Operating System Remediation System Service (OSRSS) installer is attempting an update to itself.This information helps ensure Windows is up to date.

The following fields are available:

- **InventoryVersionInstallerVersion** The versioninformation of the inventory binary generatingInstaller component. Time The system time at which the eventsevent occurred.

Microsoft.Windows.InventorySediment.GeneralServiceInstaller.InventoryMiscellaneousOfficeVBAStartSyncBinaryUpdated

DiagnosticThis event to indicate aindicates the Operating System Remediation System Service (OSRSS) updated installer binaries with new syncbinaries as part of its self-update process. This information helps ensure Windows is being generated for this objecttypeup to date.

This event includesThe following fields fromare available: MsInstallerVersion The version information of the Installer component.DeviceTime The system time at which the event occurred.DeviceInventoryChangeMicrosoft.Windows.Sediment.ServiceInstaller.Error This event indicates an error occurred in the Operating System Remediation System Service (OSRSS). The information provided helps ensure future upgrade/update attempts are more successful.

The following fields are available:

- **InventoryVersionFailureType** Thetype of error encountered. FileName The code file in which the error occurred. HResult The failure error code. InstallerVersion The versioninformation of the inventory binary generatingInstaller component. LineNumber The line number in the eventscode file at which the error occurred.Time The system time at which the event occurred.

Microsoft.Windows.InventorySediment.GeneralServiceInstaller.InventoryMiscellaneousUUPInfoAddInstallerLaunched

Provides data on Unified Update PlatformThis event indicates the Operating System Remediation System Service (UUPOSRSS) products and what version they are athas launched.The information provided helps ensure Windows is up to date.

This event includesThe following fields fromare available: MsInstallerVersion The version information of the Installer component.DeviceTime The system time at which the event occurred.DeviceInventoryChangeMicrosoft.Windows.Sediment.ServiceInstaller.ServiceInstalled T

his event indicates the Operating System Remediation System Service (OSRSS) successfully installed the Installer Component.This information helps ensure Windows is up to date.

The following fields are available:

- **Identifier** UUP identifier
- **LastActivatedVersion** Last activatedInstallerVersion The version
- **PreviousVersion** Previous version
- **Source** UUP source
- **Version** UUP versioninformation of the Installer component. Time The system time at which the event occurred.

Microsoft.Windows.InventorySediment.GeneralServiceInstaller.InventoryMiscellaneousUUPInfoRemoveServiceRestarted

Indicates that this particular data object represented byThis event indicates the objectInstanceIdOperating System Remediation System Service (OSRSS) has restarted after installing an updated version of itself. This information helps ensure Windows is nolonger presentup to date.

This event includesThe following fields fromare available: MsInstallerVersion The version information of the Installer component.DeviceTime The system time at which the event occurred.DeviceInventoryChange.

Microsoft.Windows.InventorySediment.GeneralServiceInstaller.InventoryMiscellaneousUUPInfoStartSyncServiceStarted

DiagnosticThis eventindicates the Operating System Remediation System Service (OSRSS) has started after installing an updated version of itself. This information helps ensure Windows is up to indicate a new sync is being generated for this object typedate.

This event includesThe following fields fromare available: MsInstallerVersion The version information of the Installer component.DeviceTime The system time at which the event occurred.DeviceInventoryChange.

Microsoft.Windows.InventorySediment.IndicatorsServiceInstaller.ChecksumServiceStopped

This
event summarizesindicates the counts for the InventoryMiscellaneousUexIndicatorAdd eventsOperating System Remediation System Service (OSRSS) was stopped by a self-updated to install an updated version of itself.This information helps ensure Windows is up to date.

The following fields are available:

- **ChecksumDictionary** A countInstallerVersion The version information of each operatingthe Installer component. Time Thesystem indicator.
- **PCFP** Equivalent totime at which the InventoryId field that is found in other core eventsevent occurred.

Microsoft.Windows.InventorySediment.IndicatorsServiceInstaller.InventoryMiscellaneousUexIndicatorAddUninstallerCompleted

These events representThis event indicates the basic metadata aboutOperating System Remediation System Service (OSRSS) successfully uninstalled the OS indicators installed on the system which are used for keeping the deviceversion as part of a self-update. This information helps ensure Windows is up to date.

This event includesThe following fields fromare available: MsInstallerVersion The version information of the Installer component.DeviceTime The system time at which the event occurred.DeviceInventoryChangeMicrosoft.Windows.Sediment.ServiceInstaller.UninstallerLaunched This event indicates the Operating System Remediation System Service (OSRSS) successfully started the Uninstaller as part of a self-update.This information helps ensure Windows is up to date.

The following fields are available:

- **IndicatorValueInstallerVersion** The indicator valueversion information of the Installer component.Time The system time at which the event occurred.

Microsoft.Windows.InventorySediment.IndicatorsServiceInstaller.InventoryMiscellaneousUexIndicatorRemoveUpdaterCompleted

This event is a counterpart to InventoryMiscellaneousUexIndicatorAdd that indicates that the item has been removedOperating System Remediation System Service (OSRSS) successfully completed the self-update operation.This information helps ensure Windows is up to date.

This event includesThe following fields fromare available: MsInstallerVersion The version information of the Installer component.DeviceTime The system time at which the event occurred.DeviceInventoryChange.

Microsoft.Windows.InventorySediment.IndicatorsServiceInstaller.InventoryMiscellaneousUexIndicatorStartSyncUpdaterLaunched

This event indicates that a new setthe Operating System Remediation System Service (OSRSS) successfully launched the self-updater after downloading it. This information helps ensure Windows is up to date. The following fields are available: InstallerVersion The version information of InventoryMiscellaneousUexIndicatorAdd events will be sentthe Installer component.Time The system time at which the event occurred. Microsoft.Windows.SedimentLauncher.Applicable
Indicates whether a given plugin is applicable.

This event includesThe following fields fromare available: MsCV Correlation vector.DeviceDetectedCondition Boolean true if detect condition is true and perform action will be run.DeviceInventoryChangeGlobalEventCounter Client side counter which indicates ordering of events sent by this user. IsSelfUpdateEnabledInOneSettings True if self update enabled in Settings. IsSelfUpdateNeeded True if self update needed by device. PackageVersion Current package version of Remediation. PluginName Name of the plugin specified for each generic plugin event. Result This is the HRESULT for detection or perform action phases of the plugin. Microsoft.Windows.SedimentLauncher.Completed Indicates whether a given plugin has completed its work.
The following fields are available:

KernelCV Correlation vector. FailedReasons Concatenated list of failure reasons. GlobalEventCounter Client side counter which indicates ordering of eventssent by this user. PackageVersion Current package version of Remediation. PluginName Name of the plugin specified for each generic plugin event. Result This is the HRESULT for detection or perform action phases of the plugin. SedLauncherExecutionResult HRESULT for one execution of the Sediment Launcher. IOMicrosoft.Windows.SedimentLauncher.Error

This event indicatesan error occurred during the numberexecution of bytes read from or read by the OS and written to or written bythe OS upon system startupplug-in.The information provided helps ensure future upgrade/update attempts are more successful.

The following fields are available:

- **BytesReadHResult** The total numberresult for the Detection or Perform Action phases of bytes read from or read by the OSupon system startupplug-in.
- **BytesWrittenMessage A message containing information about the error that occurred (if any). PackageVersion** The total version number of bytes written to or written by the OS upon system startupcurrent remediation package.

Microsoft.Windows.KernelSedimentLauncher.BootEnvironmentFallbackError This event indicates that an error occurred during execution of the plug-in fallback.OsLaunchThe following fields are available: s0 Error occurred during execution of the plugin fallback. See Microsoft.Windows.SedimentLauncher.wilResult . Microsoft.Windows.SedimentLauncher.Information

OSThis event provides general information collected during Boot, used to evaluatereturned from the successplug-in. The following fields are available: HResult This is the HRESULT for detection or perform action phases of

the upgrade processplugin.Message Information message returned from a plugin containing only information internal to the plugins execution. PackageVersion Current package version of Remediation. Microsoft.Windows.SedimentLauncher.Started
This event indicates that a given plug-in has started.

The following fields are available:

- **BootApplicationId**CV Correlation vector. GlobalEventCounter Client side counter which indicates ordering of events sent by this user. PackageVersion Current package version of Remediation. PluginName Name of the plugin specified for each generic plugin event. Result This field tells us whatis the OS Loader Application Identifier isHRESULT for detection or perform action phases of the plugin.
- **BootAttemptCount**Microsoft.Windows.SedimentLauncher.wilResult This event provides the result from the Windows internal library. The numberfollowing fields are available: callContext List of consecutive timestelemetry activities containing this error. currentContextId Identifier for the boot manager has attemptednewest telemetry activity containing this error. currentContextMessage Custom message associated with the newest telemetry activity containing this error (if any). currentContextName Name of the newest telemetry activity containing this error. failureCount Number of failures seen within the binary where the error occurred. failureId Identifier assigned to boot into this operating systemfailure.
- **BootSequence** The current Boot IDfailureType Indicates what type of failure was observed (exception, used to correlate eventsrelated to a particular boot sessionreturned error, logged error or fail fast). fileName Source code file name where the error occurred.
- **BootStatusPolicy** Identifiesfunction Name of the applicable Boot Status Policyfunction where the error occurred.
- **BootType** Identifieshresult Failure error code. lineNumber Line number within the typesource code file where the error occurred. message Custom message associated with the failure (if any). module Name of bootthe binary where the error occurred. originatingContextId Identifier for the oldest telemetry activity containing this error. originatingContextMessage Custom message associated with the oldest telemetry activity containing this error (eif any). originatingContextName Name of the oldest telemetry activity containing this error.g.: "Cold", "Hiber", "Resume").
- **EventTimestamp** Seconds elapsed since an arbitrary time pointthreadId Identifier of the thread the error occurred on.Microsoft.Windows.SedimentService.Applicable This can be usedevent indicates whether a given plug-in is applicable. The following fields are available: CV Correlation vector. DetectedCondition Determine whether action needs to identify the timedifference in successive boot attempts being maderun based on device properties.
- **FirmwareResetReasonEmbeddedController** Reason for system reset providedGlobalEventCounter Client side counter which indicates ordering of events sent by firmwarethis user.
- **FirmwareResetReasonEmbeddedControllerAdditional** Additional information on system reset reason provided byfirmwareIsSelfUpdateEnabledInOneSettings Indicates ifself update is enabled in One Settings. IsSelfUpdateNeeded Indicates if self update is needed.
- **FirmwareResetReasonPch** ReasonPackageVersion Current package version of Remediation. PluginName Name of the plugin. Result This is the HRESULT for system reset provideddetection or perform action phases of the plugin. Microsoft.Windows.SedimentService.Completed This event indicates whether a given plug-in has completed its work. The following fields are available: CV Correlation vector. FailedReasons List of reasons when the plugin action failed. GlobalEventCounter Client side counter which indicates ordering of events sent by firmwarethis user.
- **FirmwareResetReasonPchAdditional** Additional information on system reset reason providedPackageVersion Current package version of Remediation. PluginName Name of the plugin specified for each generic plugin event. Result This is the HRESULT for detection or perform action phases of the plugin. SedimentServiceCheckTaskFunctional True/False if scheduled task check succeeded. SedimentServiceCurrentBytes Number of current private bytes of memory consumed by firmwaresedsvc.exe. SedimentServiceKillService True/False if neededservice is marked for kill (Shell.
- **FirmwareResetReasonSupplied** Flag indicating that a reasonKillService). SedimentServiceMaximumBytes Maximum bytes allowed for system reset was provided by firmwarethe service.
- **IO** AmountSedimentServiceRetrievedKillService True/False if result of data written to and read fromOne Settings check for kill succeeded – we only send back one of these indicators (not for each call). SedimentServiceStopping True/False indicating whether the disk byservice is stopping. SedimentServiceTaskFunctional True/False if scheduled task is functional. If task is not functional this indicates plugins will be run. SedimentServiceTotalIterations Number of 5 second iterations service will wait before running again. Microsoft.Windows.SedimentService.Error This event indicates whether an error condition occurred inthe OS Loader during boot plug-in. See IOThe following fields are available: HResult This is the HRESULT for detection or perform action phases of the plugin.
- **LastBootSucceeded** Flag indicating whetherMessage Custom message associated with the last boot was successfulfailure (if any). PackageVersion Current package version of Remediation.
- **LastShutdownSucceeded** Flag indicatingMicrosoft.Windows.SedimentService.FallbackError This event indicates whetheran error occurred for a fallback in the last shutdown was successfulplug-in.

- **MaxAbove4GbFreeRange** ~~**The following fields are available: s0 Event returned when an error occurs for a fallback in the plugin. See Microsoft.Windows.SedimentService.wilResult . Microsoft.Windows.SedimentService.Information**~~ This fielddes cribes~~event provides general information returned from~~ the largest memory range~~plug-in. The following fields are~~available above 4Gb~~: HResult This is the HRESULT for detection or perform action phases of the plugin~~.
- **MaxBelow4GbFreeRange** ~~**Message Custom message associated with the failure (if any). PackageVersion Current package version of Remediation. Microsoft.Windows.SedimentService.Started**~~ This field describes~~event indicates a specified plug-in has started. This information helps ensure Windows is up to date. The following fields are available: CV The Correlation Vector. GlobalEventCounter The client-side counter that indicates ordering of events. PackageVersion The version number of~~ the largest memory range~~current remediation package. PluginName Name of the plugin specified for each generic plugin event. Result This is the HRESULT for Detection or Perform Action phases of the plugin. Microsoft.Windows.SedimentService.wilResult This event provides the result from the Windows internal library. The following fields are~~ available below 4Gb~~: callContext List of telemetry activities containing this error~~.
- **MeasuredLaunchPrepared** This field tells us~~currentContextId Identifier for the newest telemetry activity containing this error. currentContextMessage Custom message associated with the newest telemetry activity containing this error (~~ if~~any). currentContextName Name of~~ the OS launch~~newest telemetry activity containing this error. failureCount Number of failures seen within the binary where the error occurred. failureId Identifier assigned to this failure. failureType Indicates what type of failure~~ was initiated using Measured/Secure Boot over DRTM~~observed~~ (Dynamic Root~~exception, returned error, logged error or fail fast). fileName Source code file name where the error occurred. function Name~~ of Trust for Measurement~~the function where the error occurred. hresult Failure error code. lineNumber Line number within the source code file where the error occurred. message Custom message associated with the failure (if any~~).
- **MeasuredLaunchResume** This field tells us if Dynamic Root~~module Name~~ of Trust~~the binary where the error occurred. originatingContextId Identifier~~ for Measurement~~the oldest telemetry activity containing this error. originatingContextMessage Custom message associated with the oldest telemetry activity containing this error~~ (DRTM) was used when resuming from~~hibernation~~if any). originatingContextName Name of the oldest telemetry activity containing this error~~.
- **MenuPolicy** Type~~threadId Identifier~~ of advanced options menu~~the thread the error occurred on. Setup events SetupPlatformTel. SetupPlatformTelActivityEvent This event sends basic metadata about the SetupPlatform update installation process, to help keep Windows up to date. The following fields are available: ActivityId Provides a unique Id to correlate events~~ that should be shown ~~occur between a activity start event, and a stop event ActivityName Provides a friendly name of the package type that belongs~~ to the user~~ActivityId~~ (Legacy~~Setup~~, Standard~~LanguagePack, GDR, Driver,~~ etc.~~).~~
- **RecoveryEnabled** Indicates whether recovery is enabled~~.) FieldName Retrieves the event name/data point~~.
- **SecureLaunchPrepared** ~~**Examples: InstallStartTime, InstallEndtime, OverallResult etc. GroupName Retrieves the groupname the event belongs to. Example: Install Information, DU Information, Disk Space Information etc. value Value associated with the corresponding event name. For example, time-related events will include the system time Value Value associated with the corresponding event name. For example, time-related events will include the system time SetupPlatformTel.SetupPlatformTelActivityStarted**~~ This field indicates if DRTM was prepared during boot~~event sends basic metadata about the update installation process generated by SetupPlatform to help keep Windows up to date~~.
- **TcbLaunch** Indicates whether~~The following fields are available: Name The name of~~ the Trusted Computing Base was used~~during dynamic update type. Example: GDR driver SetupPlatformTel.SetupPlatformTelActivityStopped This event sends basic metadata about~~ the boot flow~~update installation process generated by SetupPlatform to help keep Windows up to date~~.
- **UserInputTime** ~~**SetupPlatformTel.SetupPlatformTelEvent This service retrieves events generated by SetupPlatform, the engine that drives the various deployment scenarios.**~~ The amount of~~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 (Field Name). For example: For~~ time~~related events this will include~~ the loaderapplication spent waiting for user input~~system time~~.

Privacy consent logging~~Shared PC~~ events

Microsoft.Windows.Shell~~SharedPC~~.PrivacyConsentLogging~~AccountManager~~.PrivacyConsentCompleted~~DeleteUserAccount~~

This event is used to determine whether the~~Activity for deletion of a~~ user successfully completed~~account for devices set up for Shared PC mode as part of~~ the privacy consent experience~~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:

- **presentationVersion** Which display versionaccountType The type of the privacy consent experience the user completed
- **privacyConsentStateaccount that was deleted. Example: AD, AAD, or Local userSid** The current statesecurity identifier of the privacy consent experience
- **settingsVersion** Which setting version of the privacy consent experience theaccount. wilActivity Windows Error Reporting data collected when there is a failure in deleting a user completed
- **userOobeExitReason** The exit reason ofaccount with the privacy consent experienceTransient Account Manager. See wilActivity .

Microsoft.Windows.ShellSharedPC.PrivacyConsentLoggingAccountManager.PrivacyConsentStatusSinglePolicyEvaluation

Event tells us effectivenessActivity for run of new privacy experiencethe 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:

- **isAdmin** whetherevaluationTrigger When was the person whoTransient Account Manager policies ran? Example: At log off or during maintenance hours totalAccountCount The number of accounts on a device after running the Transient Account Manager policies. wilActivity Windows Error Reporting data collected when there is loggingfailure in is an admin
- **isExistingUser** whetherevaluating accounts to be deleted with the account existedTransient Account Manager. See wilActivity . wilActivity This event provides a Windows Internal Library context used for Product and Service diagnostics. The following fields are available: callContext The function where the failure occurred. currentContextId The ID of the current call context where the failure occurred. currentContextMessage The message of the current call context where the failure occurred. currentContextName The name of the current call context where the failure occurred. failureCount The number of failures for this failure ID. failureId The ID of the failure that occurred. failureType The type of the failure that occurred. fileName The file name where the failure occurred. function The function where the failure occurred. hresult The HResult of the overall activity. lineNumber The line number where the failure occurred. message The message of the failure that occurred. module The module where the failure occurred. originatingContextId The ID of the originating call context that resulted inthe failure. originatingContextMessage The message of the originating call context that resulted in the failure. originatingContextName The name of the originating call context that resulted in the failure. threadId The ID of the thread on which the activity is executing. wilResult This event provides a downlevel OS
- **isLaunching** Whether or notWindows Internal Library context used for Product and Service diagnostics. The following fields are available: callContext The call context stack where failure occurred. currentContextId The ID of the privacy consent experiencewill be launched
- **isSilentElevation** whethercurrent call context where the user has most restrictive UAC controls
- **privacyConsentState** whetherfailure occurred. currentContextMessage The message of the user has completed privacyexperience
- **userRegionCodecurrent call context where the failure occurred. currentContextName** Thename of the current user's region settingcall context where the failure occurred. failureCount The number of failures for this failure ID. failureId The ID of the failure that occurred. failureType The type of the failure that occurred. fileName The file name where the failure occurred. function The function where the failure occurred. hresult The HResult of the overall activity. lineNumber The line number where the failure occurred. message The message of the failure that occurred. module The module where the failure occurred. originatingContextId The ID of the originating call context that resulted in the failure. originatingContextMessage The message of the originating call context that resulted in the failure. originatingContextName The name of the originating call context that resulted in the failure. threadId The ID of the thread on which the activity is executing.

Software update events

SoftwareUpdateClientTelemetry.CheckForUpdates

Scan processThis event on Windows Updatesends tracking data about the software distribution client. See the EventScenariofieldcheck for specifics (started/failed/succeeded).content that is applicable to a device, to help keep Windows up to date

The following fields are available:

- **ActivityMatchingId** Contains a unique ID identifying a single CheckForUpdates session from initialization to completion.
- **AllowCachedResults** Indicates if the scan allowed using cached results.
- **ApplicableUpdateInfo** Metadata for the updates which were detected as applicable

- **BiosFamily** The family of the BIOS (Basic Input Output System).
- **BiosName** The name of the device BIOS.
- **BiosReleaseDate** The release date of the device BIOS.
- **BiosSKUNumber** The sku number of the device BIOS.
- **BIOSVendor** The vendor of the BIOS.
- **BiosVersion** The version of the BIOS.
- **BranchReadinessLevel** The servicing branch configured on the device.
- **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.
- **CallerApplicationName** The name provided by the caller who initiated API calls into the software distribution client.
- **CapabilityDetectoidGuid** The GUID for a hardware applicability detectoid that could not be evaluated.
- **CDNCountryCode** Two letter country abbreviation for the Content Distribution Network (CDN) location.
- **CDNId** The unique identifier of a specific device, used to identify how many devices are encountering success or a particular issue.
- **ClientVersion** The version number of the software distribution client.
- **CommonProps** A bitmask for future flags associated with the Windows Update client behavior. No data is currently reported int his field. Expected value for this field is 0.
- **Context** Gives context on where the error has occurred. Example: AutoEnable, GetSLSData, AddService, Misc, or Unknown
- **CurrentMobileOperator** The mobile operator the device is currently connected to.
- **DeferralPolicySources** Sources for any update deferral policies defined (GPO = 0x10, MDM = 0x100, Flight = 0x1000, UX = 0x10000).
- **DeferredUpdates** Update IDs which are currently being deferred until a later time
- **DeviceModel** What is the device model.
- **DriverError** The error code hit during a driver scan. This is 0 if no error was encountered.
- **DriverExclusionPolicy** Indicates if the policy for not including drivers with Windows Update is enabled.
- **DriverSyncPassPerformed** Were drivers scanned this time?
- **EventInstanceID** A globally unique identifier for event instance.
- **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.
- **ExtendedMetadataCabUrl** Hostname that is used to download an update.
- **ExtendedStatusCode** Secondary error code for certain scenarios where StatusCode wasn't specific enough.
- **FailedUpdateGuids** The GUIDs for the updates that failed to be evaluated during the scan.
- **FailedUpdatesCount** The number of updates that failed to be evaluated during the scan.
- **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).
- **FlightBranch** The branch that a device is on if participating in flighting (pre-release builds).
- **FlightRing** The ring (speed of getting builds) that a device is on if participating in flighting (pre-release builds).
- **HomeMobileOperator** The mobile operator that the device was originally intended to work with.
- **IntentPFNs** Intended application-set metadata for atomic update scenarios.
- **IPVersion** Indicates whether the download took place over IPv4 or IPv6
- **IsWUfBDualScanEnabled** Indicates if Windows Update for Business dual scan is enabled on the device.
- **IsWUfBEnabled** Indicates if Windows Update for Business is enabled on the device.
- **IsWUfBFederatedScanDisabled** Indicates if Windows Update for Business federated scan is disabled on the device.
- **MetadataIntegrityMode** The mode of the update transport metadata integrity check. 0-Unknown, 1-Ignoe, 2-Audit, 3-Enforce
- **MSIError** The last error that was encountered during a scan for updates.
- **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
- **NumberOfApplicationsCategoryScanEvaluated** The number of categories (apps) for which an app update scan checked
- **NumberOfLoop** The number of round trips the scan required
- **NumberOfNewUpdatesFromServiceSync** The number of updates which were seen for the first time in this scan
- **NumberOfUpdatesEvaluated** The total number of updates which were evaluated as a part of the scan
- **NumFailedMetadataSignatures** The number of metadata signatures checks which failed for new metadata synced down.
- **Online** Indicates if this was an online scan.
- **PausedUpdates** A list of UpdateIds which that currently being paused.

- **PauseFeatureUpdatesEndTime** If feature 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.
- **PauseQualityUpdatesEndTime** If quality OS updates are paused on the device, this is the date and time for the end of the pause time window.
- **PauseQualityUpdatesStartTime** If quality OS updates are paused on the device, this is the date and time for the beginning of the pause time window.
- **PhonePreviewEnabled** Indicates whether a phone was getting preview build, prior to flighting (pre-release builds) being introduced.
- **ProcessName** The process name of the caller who initiated API calls, in the event where CallerApplicationName was not provided.
- **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).
- **RelatedCV** The previous Correlation Vector that was used before swapping with a new one
- **ScanDurationInSeconds** The number of seconds a scan took
- **ScanEnqueueTime** The number of seconds it took to initialize a scan
- **ScanProps** This is a 32-bit integer containing Boolean properties for a given Windows Update scan. The following bits areused; all remaining bits are reserved and set to zero. Bit 0 (0x1): IsInteractive - is set to 1 if the scan is requested by a user, or 0if the scan is requested by Automatic Updates. Bit 1 (0x2): IsSeeker - is set to 1 if the Windows Update client's Seekerfunctionality is enabled. Seeker functionality is enabled on certain interactive scans, and results in the scans returning certainupdates that are in the initial stages of release (not yet released for full adoption via Automatic Updates).
- **ServiceGuid** An ID which represents which service the software distribution client is checking for content (Windows Update, Microsoft~~Windows~~ Store, etc.).
- **ServiceUrl** The environment URL a device is configured to scan with
- **ShippingMobileOperator** The mobile operator that a device shipped on.
- **StatusCode** Indicates the result of a CheckForUpdates event (success, cancellation, failure code HResult).
- **SyncType** Describes the type of scan the event was
- **SystemBIOSMajorRelease** Major version of the BIOS.
- **SystemBIOSMinorRelease** Minor version of the BIOS.
- **TargetMetadataVersion** For self-initiated healing, this is the target version of the SIH engine to download (if needed). If not, the value is null.
- **TotalNumMetadataSignatures** The total number of metadata signatures checks done for new metadata that was synced down.
- **WebServiceRetryMethods** Web service method requests that needed to be retried to complete operation.
- **WUDeviceID** The unique identifier of a specific device, used to identify how many devices are encountering success or a particular issue.

SoftwareUpdateClientTelemetry.Download~~Commit~~

Download process~~This~~ event for target update~~sends data~~ on~~whether the Update Service has been called to execute an upgrade, to help keep~~ Windows Update~~up to date. The following fields are available: BiosFamily The family of the BIOS (Basic Input Output System). BiosName The name of the device BIOS. BiosReleaseDate The release date of the device BIOS. BiosSKUNumber The sku number of the device BIOS. BIOSVendor The vendor of the BIOS. BiosVersion The version of the BIOS. 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. CallerApplicationName The name provided by the caller who initiated API calls into the software distribution client~~ client~~ClientVersion The version number of the software distribution client.~~ See~~DeviceModel What is the~~device model. EventInstanceID A globally unique identifier for event instance.~~ EventScenario field~~State of call EventType Possible values are "Child", "Bundle", or "Driver". FlightId The specific id of the flight the device is getting HandlerType Indicates the kind of content (app, driver, windows patch, etc.) RevisionNumber Unique revision number of Update ServerId Identifier~~ for specifics(started/failed/succeeded).~~the service to which the software distribution client is connecting, such as Windows Update and Windows Store. SystemBIOSMajorRelease Major version of the BIOS. SystemBIOSMinorRelease Minor version of the BIOS. UpdateId Unique Update ID WUDeviceID UniqueDeviceID 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:

- **ActiveDownloadTime** Number ofHow long the download took, in seconds, excluding time where the update waswasn'tactively being downloaded.
- **AppXBlockHashFailures** Indicates the number of blocks that failed hash validation during download of the app payload.
- **AppXBlockHashValidationFailureCount** A count of the number of blocks that have failed validation after being downloaded.
- **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.
- **AppXScope** Indicates the scope of the app download. The values can be one of the following: "RequiredContentOnly" - onlythe content required to launch the app is being downloaded; "AutomaticContentOnly" - only the optional [automatic] contentfor the app (the ones that can downloaded after the app has been launched) is being downloaded; "AllContent" - all contentfor the app , including the optional [automatic] content, is being downloaded.
- **BiosFamily** The family of the BIOS (Basic Input Output System).
- **BiosName** The name of the device BIOS.
- **BiosReleaseDate** The release date of the device BIOS.
- **BiosSKUNumber** The sku number of the device BIOS.
- **BIOSVendor** The vendor of the BIOS.
- **BiosVersion** The version of the BIOS.
- **BundleBytesDownloaded** Number ofHow many bytes were downloaded for the specific content bundle.
- **BundleId** Identifier associated with the specific content bundle; should not be all zeros if the bundleID was found.
- **BundleRepeatFailCount** Indicates whether this particular update bundle has previously failed.
- **BundleRepeatFailFlag** Indicates whether this particular update bundlehad previously failed to download.
- **BundleRevisionNumber** Identifies the revision number of the content bundle.
- **BytesDownloaded** Number ofHow many bytes that were downloaded for an individual piece of content (not the entire bundle).
- **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.
- **CallerApplicationName** The name provided by the caller who initiated API calls into the software distribution client.
- **CbsDownloadMethod** Indicates whether the download was a full-file download or a partial/delta download.
- **CbsMethod** The method used for downloading the update content related to the Component Based Servicing (CBS)technology. This value can be one of the following: (1) express download method was used for download; (2) SelfContaineddownload metho d was used for download indicating the update had no express content; (3) SelfContained download methodwas used indicating that the update has an express payload, but the server is not hosting it; (4) SelfContained downloadmethod was used indicating t hat range requests are not supported; (5) SelfContained download method was used indicatingthat the system does not support express download (dpx.dll is not present); (6) SelfContained download method was usedindicating that self-contained download method was selected previously; (7) SelfContained download method was usedindicating a fall back to self-contained if the number of requests made by DPX exceeds a certain threshold.
- **CDNCountryCode** Two letter country abbreviation for the Content Distribution Network (CDN) location.
- **CDNId** ID which defines which CDN the software distribution client downloaded the content from.ClientManagedByWSUSServer Indicates whether the client is managed by Windows Server Update Services (WSUS).
- **ClientVersion** The version number of the software distribution client.
- **CommonProps** A bitmask for future flags associated with the Windows Update client behavior. No value is currently reportedin t his field. Expected value for this field is 0.
- **ConnectTime** Indicates the cumulative sum (in seconds) of the time it took to establish the connection for all updates in anupda te bundle.
- **CurrentMobileOperator** The mobile operator the device is currently connected to.
- **DeviceModel** What is the device model.DeviceOEM What OEM does this device belong to.
- **DownloadPriority** Indicates whether a download happened at background, normal, or foreground priority.
- **DownloadProps** IndicatesDownloadScenarioId A unique ID for a bitmask forgiven download operations indicating: (1) if anupda te was downloadedused to a system volume (least significant bit itie together WU and DO events.e. bit 0); (2) if theupdate was fr om a channel other than the installed channel (bit 1); (3) if the update was for a product pinned by policy (bit 2);(4) if the deploy ment action for the update is uninstall (bit 3).
- **DownloadType** Differentiates the download type of SIH downloads between Metadata and Payload downloads.Edition Indicates the edition of Windows being used.
- **EventInstanceID** A globally unique identifier for event instance.EventNamespaceID Indicates whether the event succeeded or failed. Has the format EventType+Event where Event is Su cceeded, Cancelled, Failed, etc.
- **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.

- **EventType** Possible values are Child, Bundle, or Driver.
- **ExtendedStatusCode** Secondary error code for certain scenarios where StatusCode wasn't specific enough.
- **FeatureUpdatePause** Indicates whether feature OS updates are paused on the device.
- **FlightBranch** The branch that a device is on if participating in flighting (pre-release builds).
- **FlightBuildNumber** If this download was for a flight (pre-release build), this indicates the build number of that flight.
- **FlightId** The specific IDid of the flight (pre-release build) the device is getting.
- **FlightRing** The ring (speed of getting builds) that a device is on if participating in flighting (pre-release builds).
- **HandlerType** Indicates what kind of content is being downloaded (app, driver, windows patch, etc.).
- **HardwareId** If this download was for a driver targeted to a particular device model, this ID indicates the model of the device.
- **HomeMobileOperator** The mobile operator that the device was originally intended to work with.
- **HostName** The hostname URL the content is downloading from.
- **IPVersion** Indicates whether the download took place over IPv4 or IPv6.~~IsAOACDevice Is it Always On, Always Connected?~~
- **IsDependentSet** Indicates whether a driver is a part of a larger System Hardware/Firmware Update
- **IsWUfBDualScanEnabled** Indicates if Windows Update for Business dual scan is enabled on the device.
- **IsWUfBEnabled** Indicates if Windows Update for Business is enabled on the device.
- **NetworkCost** A flag indicating the cost of the network used for downloading the update content. The values can be: 0x0(Unkown); 0x1 (Network cost is unrestricted); 0x2 (Network cost is fixed); 0x4 (Network cost is variable); 0x10000 (Network costover data limit); 0x20000 (Network cost congested); 0x40000 (Network cost roaming); 0x80000 (Network cost approachingdata limit).
- **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."
- **PackageFullName** The package name of the content.
- **PhonePreviewEnabled** Indicates whether a phone was opted-in to getting preview builds, prior to flighting (pre-release builds) being introduced.
- **PostDnldTime** Time taken (in seconds) to signal download completion after the last job has completed downloadingpayload.~~PlatformRole The PowerPlatformRole as defined on MSDN~~
- **ProcessName** The process name of the caller who initiated API calls, in the event where CallerApplicationName was not provided.~~ProcessorArchitecture Processor architecture of the system (x86, AMD64, ARM).~~
- **QualityUpdatePause** Indicates whether quality OS updates are paused on the device.
- **Reason** A 32-bit integer representing the reason the update is blocked from being downloaded in the background.
- **RegulationReason** The reason that the update is regulated
- **RegulationResult** The result code (HResult) of the last attempt to contact the regulation web service for download regulationof update content.
- **RelatedCV** The previous Correlation Vector that was used before swapping with a new one.
- **RepeatFailCount** Indicates whether this specific piece of content has previously failed.
- **RepeatFailFlag** Indicates whether this specific piece of content had previously failed to download.
- **RevisionNumber** Identifies the revision number of this specific piece of content.
- **ServiceGuid** An ID thatwhich represents which service the software distribution client is installing content for (Windows Update, Microsoft Store, etc.).
- **Setup360Phase** If the download is for an operating system upgrade, this datapoint indicates which phase of the upgrade is underway.
- **ShippingMobileOperator** The mobile operator that a device shipped on.
- **SizeCalcTime** Time taken (in seconds) to calculate the total download size of the payload.
- **StatusCode** Indicates the result of a Download event (success, cancellation, failure code HResult).
- **SystemBIOSMajorRelease** Major version of the BIOS.
- **SystemBIOSMinorRelease** Minor version of the BIOS.
- **TargetGroupId** For drivers targeted to a specific device model, this ID indicates the distribution group of devices receiving that driver.
- **TargetingVersion** For drivers targeted to a specific device model, this is the version number of the drivers being distributed to the device.
- **TargetMetadataVersion** For self-initiated healing, this is the target version of the SIH engine to download (if needed). If not, the value is null.
- **ThrottlingServiceHResult** Result code (success/failure) while contacting a web service to determine whether this device should download content yet.
- **TimeToEstablishConnection** Time (in ms) it took to establish the connection prior to beginning downloaded.
- **TotalExpectedBytes** The total count of bytes that the download is expected to be.
- **UpdateId** An identifier associated with the specific piece of content.

- **UpdateID** An identifier associated with the specific piece of content.
- **UpdateImportance** Indicates whether a piece of content was marked as Important, Recommended, or Optional.
- **UsedDO** Whether the download used the delivery optimization service.
- **UsedSystemVolume** Indicates whether the content was downloaded to the device's main system storage drive, or an alternate storage drive.
- **WUDeviceID** The unique identifier of a specific device, used to identify how many devices are encountering success or a particular issue.~~WUSetting Indicates the users' current updating settings.~~

SoftwareUpdateClientTelemetry.~~DownloadCheckpoint This event provides a checkpoint between each of the Windows Update download phases for UUP content The following fields are available: CallerApplicationName The name provided by the caller who initiated API calls into the software distribution client ClientVersion The version number of the software distribution client 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 Event Type Possible values are "Child", "Bundle", "Relase" or "Driver" ExtendedStatusCode Secondary error code for certain scenarios where StatusCode wasn't specific enough FileId A hash that uniquely identifies a file FileName Name of the downloaded file FlightId The unique identifier for each flight RelatedCV The previous Correlation Vector that was used before swapping with a new one RevisionNumber Unique revision number of Update 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) UpdateId Unique Update ID WUDeviceID The unique identifier of a specific device, used to identify how many devices are encountering success or a particular issue 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:

- **BiosFamily** The family of the BIOS (Basic Input Output System).
- **BiosName** The name of the device BIOS.
- **BiosReleaseDate** The release date of the device BIOS.
- **BiosSKUNumber** The sku number of the device BIOS.
- **BIOSVendor** The vendor of the BIOS.
- **BiosVersion** The version of the BIOS.~~BundleBytesDownloaded How many bytes were downloaded for the specific content bundle?~~
- **BundleId** Identifier associated with the specific content bundle; should not be all zeros if the bundleID was found.
- **BundleRepeatFailCount** Indicates whether this particular update bundle has previously failed.
- **BundleRepeatFailFlag** Indicates whether~~Has~~ this particular update bundle previously failed to install.~~?~~
- **BundleRevisionNumber** Identifies the revision number of the content bundle.~~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.~~
- **CallerApplicationName** The name provided by the caller who initiated API calls into the software distribution client.~~CbsDownloadMethod Was the download a full download or a partial download? ClientManagedByWSUSServer Is the client managed by Windows Server Update Services (WSUS)?~~
- **ClientVersion** The version number of the software distribution client.
- **CommonProps** A bitmask for future flags associated with the Windows Update client behavior. No value is currently reportedin this field. Expected value for this field is 0.
- **CSIErrorType** The stage of CBS installation where it failed.
- **CurrentMobileOperator** The mobile~~Mobile~~ operator to which the~~that~~ device is currently connected~~to~~.
- **DeviceModel** The~~What is the~~ device model.~~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.~~
- **DriverPingBack** Contains information about the previous driver and system state.
- **DriverRecoveryIds** The list~~Edition Indicates the edition~~ of identifiers that could be~~Windows being~~ used for uninstalling thedrivers if a recovery is required.
- **EventInstanceID** A globally unique identifier for event instance.~~EventNamespaceID Indicates whether the event succeeded or failed. Has the format EventType+Event where Event is Succeeded, Cancelled, Failed, etc.~~

- **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.
- **EventType** Possible values are Child, Bundle, or Driver.
- **ExtendedErrorCode** The extended error code.
- **ExtendedStatusCode** Secondary error code for certain scenarios where StatusCode is notwasn't specific enough.
- **FeatureUpdatePause** Indicates whetherAre feature OS updates are paused on the device.?
- **FlightBranch** The branch that a device is on if participating in the Windows Insider Program.
- **FlightBuildNumber** If this installation was for a Windows Insider build, this is the build number of that build.
- **FlightId** The specific ID of the Windows Insider build the device is getting.
- **FlightRing** The ring that a device is on if participating in the Windows Insider Program.
- **HandlerType** Indicates what kind of content is being installed (for example, Example: app, driver, Windows update).
- **HardwareId** If this install was for a driver targeted to a particular device model, this ID indicates the model of the device.
- **HomeMobileOperator** The mobile operator that the device was originally intended to work with.
- **InstallProps** A bitmask for future flags associated with the install operation. No value is currently reported in this field.Expected v alue for this field is 0.
- **IntentPFNs** Intended application-set metadata for atomic update scenarios.IsAOACDevice Is it Always On, Always Connected? (Mobile device usage model)
- **IsDependentSet** Indicates whetherIs the driver is part of a larger System Hardware/Firmware update.?
- **IsFinalOutcomeEvent** Indicates whetherDoes this event signalssignal the end of the update/upgrade process.?
- **IsFirmware** Indicates whetherIs this update is a firmware update.?
- **IsSuccessFailurePostReboot** Indicates whether the update succeededDid it succeed and then failedfail after a restart.?
- **IsWUfBDualScanEnabled** Indicates whetherIs Windows Update for Business dual scan is enabled on the device.?
- **IsWUfBEnabled** Indicates whetherIs Windows Update for Business is enabled on the device.?
- **MergedUpdate** Indicates whetherWas the OS update and a BSP update merged for installation.?
- **MsiAction** The stage of MSI installation where it failed.
- **MsiProductCode** The unique identifier of the MSI installer.
- **PackageFullName** The package name of the content being installed.
- **PhonePreviewEnabled** Indicates whether a phone was getting preview build, prior to flighting being introduced.PlatformRole The PowerPlatformRole as defined on MSDN.
- **ProcessName** The process name of the caller who initiated API calls, in the event thatwhere CallerApplicationName was not provided.ProcessorArchitecture Processor architecture of the system (x86, AMD64, ARM).
- **QualityUpdatePause** Indicates whetherAre quality OS updates are paused on the device.?
- **RelatedCV** The previous Correlation Vector that was used before swapping with a new one
- **RepeatFailCountRepeatFailFlag** Indicates whether this specific piece of content hashad previously failedto install.
- **RepeatFailFlagRepeatSuccessInstallFlag** Indicates whether this specific piece of contenthad previously failed to installinstalled successful, for example if another user had already installed it.
- **RevisionNumber** The revision number of this specific piece of content.
- **ServiceGuid** An ID which represents which service the software distribution client is installing content for (Windows Update, MicrosoftWindows Store, etc.).
- **Setup360Phase** If the install is for an operating system upgrade, indicates which phase of the upgrade is underway.
- **ShippingMobileOperator** The mobile operator that a device shipped on.
- **StatusCode** Indicates the result of an installation event (success, cancellation, failure code HResult).
- **SystemBIOSMajorRelease** Major version of the BIOS.
- **SystemBIOSMinorRelease** Minor version of the BIOS.
- **TargetGroupId** For drivers targeted to a specific device model, this ID indicates the distribution group of devices receiving that driver.
- **TargetingVersion** For drivers targeted to a specific device model, this is the version number of the drivers being distributed to the device.
- **TransactionCode** The ID thatwhich represents a given MSI installation.
- **UpdateId** Unique update IDUpdateID An identifier associated with the specific piece of content.
- **UpdateImportance** Indicates whether a piece of content was marked as Important, Recommended, or Optional.
- **UsedSystemVolume** Indicates whether the content was downloaded and then installed from the device's main system storage drive, or an alternate storage drive.
- **WUDeviceID** The unique identifier of a specific device, used to identify how many devices are encountering success or a particular issue.WUSetting Indicates the user's current updating settings.

SoftwareUpdateClientTelemetry.RevertSLSDiscovery

RevertThis eventsends 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 target update on Windows Update Clienterrors. See EventScenario field for specifics (for example, Started/Failed/Succeeded).

The following fields are available:

- **BundleId** Identifier associated withEventScenario Indicates the specificpurpose of sending this event - whether because the software distribution just started checking for content bundle. Should not be all zeros if the BundleId, or whether it was found.
- **BundleRepeatFailCount**cancelled, succeeded, or failed HResult Indicates whether this particular update bundle haspreviously failed.
- **BundleRevisionNumber** Identifies the revision numberresult code of the content bundle.
- **CallerApplicationName** Name of application makingevent (success, cancellation, failure code HResult) IsBackground Indicates whether theSLS discovery event took place in the foreground or background NextExpirationTime Indicates when the SLS cab expires ServiceID An ID which represents which service the software distribution client is connecting to (Windows Update request. Used, Microsoft Store, etc.) SusClientId The unique device ID controlled by the software distribution client UrlPath Path to identify context of request.
- **ClientVersion** Versionthe SLS cab that was downloaded WUAVersion The version number of the software distribution clientSoftwareUpdateClientTelemetry.
- **CommonProps** A bitmask for future flags associated withUpdateDetected This event sends data about an AppX app that has been updated from the Windows Update client behavior. ThereMicrosoft Store, including what app needs an update and what version/architecture is no value being reportedrequired, in this field right noworder to understand and address problems with apps getting required updates. Expected valueThe following fields are available: ApplicableUpdateInfo Metadata for this fieldis 0the updates which were detected as applicable.
- **CSIErrorType** StageCallerApplicationName The name provided by the caller who initiated API calls into the software distribution client. IntentPFNs Intended application-set metadata for atomic update scenarios. NumberOfApplicableUpdates The number of CBS installation that failed.
- **DriverPingBack** Contains information aboutupdates ultimately deemed applicable to the previous driver and system stateafter the detection process is complete.
- **DriverRecoveryIds****RelatedCV** The list of identifiersprevious Correlation Vector that could bewas used for uninstallingbefore swapping with a new one. ServiceGuid An ID that represents which service the drivers if a recoverysoftware distribution clientis requiredconnecting to (Windows Update, Windows Store, etc.). WUDeviceID The unique device ID controlled by the software distribution client.
- **EventInstanceID** A globally unique identifier forSoftwareUpdateClientTelemetry.UpdateMetadataIntegrity This event instanceidentifies whether updates have been tampered with and protects against man-in-the-middle attacks.The following fields are available: EndpointUrl The endpoint URL where the device obtains update metadata. This is used to distinguish between test, staging, and production environments.
- **EventScenario** Indicates theThe purpose of thethis event (, such as scan started,scan succeeded,or scan failed, etc.).
- **EventType** Event type (Child, Bundle, Release, or Driver).
- **ExtendedStatusCode** SecondaryThe secondary status codeof the event. LeafCertId Integral ID from the FragmentSigning datafor certain scenarios where StatusCode is not specific enoughcertificate that failed.
- **FeatureUpdatePause** Indicates whether feature OS updates are paused onListOfSHA256OfIntermediateCerData A semicolon delimited list of base64 encoding of hashes for the deviceBase64CerData in the FragmentSigning data of an intermediate certificate.
- **FlightBuildNumber** IndicatesMetadataIntegrityMode The mode of the build numbertransport metadata integrity check. 0 = unknown; 1 = ignore; 2 = audit; 3 = enforce MetadataSignature A base64-encoded string of the flightsignature associated with the update metadata (specified by revision ID). RawMode The raw unparsed mode string from the SLS response.
- **FlightId****This field is null if not applicable. RawValidityWindowInDays** The specific IDraw unparsed validity window string in days of the flight the devicetimestamp token. This field is gettingnull if not applicable.
- **HandlerType** Indicates the kindRevisionId The revision ID for a specific piece of content (app, driver, windows patch, etc.).
- **HardwareId** If this download was. RevisionNumber The revision number for a driver targetedspecific piece of content. ServiceGuid Identifies the service to a particular device modelwhich the software distribution client is connected, this IDindicatesExample: Windows Update or Windows Store SHA256OfLeafCerData A base64 encoding of the modelhash for the Base64CerData in the FragmentSigning data of theleaf certificate. SHA256OfLeafCertPublicKey A base64 encoding of the hash of the Base64CertData in the FragmentSigning data of the leaf certificate. SHA256OfTimestampToken A base64-encoded string of hash of the timestamp token blob. SignatureAlgorithm The hash algorithm for the metadata signature. SLSPrograms A test program to which a devicemay have opted in.

- **IsFinalOutcomeEvent** Indicates whether thisExample: Insider Fast StatusCode The status code of the event signals the end. TimestampTokenCertThumbprint The thumbprint of the update/upgrade processencoded timestamp token.
- **IsFirmware** Indicates whether an updateTimestampTokenId The time this wascreated. It is encoded in a firmwaretimestamp blob and will be zero if the token is malformed. UpdateId The updateID for a specific piece of content.
- **IsSuccessFailurePostReboot** Indicates whether an initial successValidityWindowInDays The validity window that's in effect when verifying the timestamp. Update Assistant events Microsoft.Windows.UpdateAssistant.Orchestrator.BlockingEventId The event sends basic info on the reason that Windows 10 was a failure after a rebootnot updated due to compatibility issues, previous rollbacks, or admin policies.
- **IsWUfBDualScanEnabled** Flag indicating whether WU-for-Business dual scanThe following fields are available: ApplicabilityBlockedReason Blocked due to an applicability issue. BlockWuUpgrades The upgrade assistant is enabledoncurrently blocked. clientID An identification of thecurrent release of Update Assistant. CloverTrail This deviceis Clovertrail.
- **IsWUfBEnabled** Flag indicating whether WU-for-BusinessDeviceIsMdmManaged This device is enabled onMDM managed. IsNetworkAvailable If the devicenetwork is not available.
- **MergedUpdate** Indicates whetherIsNetworkMetered If network is metered. IsSccmManaged This device is SCCM managed. NewlyInstalledOs OS is newly installed quiet period. PausedByPolicy Updates are paused by policy. RecoveredFromRS3 Previously recovered from RS3. RS1UninstallActive Blocked due to an OS update andactive RS1 uninstall. RS3RollBacks Exceeded number of allowable RS3 rollbacks. triggerTaskSource Describe which task launches this instance. WsusManaged This device is WSUS managed. ZeroExhaust This device is zero exhaust. Microsoft.Windows.UpdateAssistant.Orchestrator.DeniedLaunchEventId The event sends basic info when a BSP update weremerged for installdevice was blocked or prevented from updating to the latest Windows 10 version.
- **ProcessName** Process nameThe following fields are available: clientID An identification of the caller who initiated API callsintocurrent release of Update Assistant. denyReason All the software distribution clientreasons why the Update Assistant was prevented from launching.
- **QualityUpdatePause** Indicates whether quality OS updatesBitmask with values from UpdateAssistant.cpp eUpgradeModeReason. triggerTaskSource Describe which task launches this instance. Microsoft.Windows.UpdateAssistant.Orchestrator.FailedLaunchEventId Event to mark that Update Assistant Orchestrator failed to launch Update Assistant. The following fields are pausedavailable: clientID An identification of the current release of Update Assistant. hResult Error code of the Update Assistant Orchestrator failure. triggerTaskSource Describe which task launches this instance. Microsoft.Windows.UpdateAssistant.Orchestrator.FailedOneSettingsQueryEventId Event indicating One Settings was not queried by update assistant. The following fields are available: clientID An identification of the current release of Update Assistant. hResult Error code of One Settings query failure. Microsoft.Windows.UpdateAssistant.Orchestrator.LaunchEventId This event sends basic information onwhether the deviceshould be updated to the latest Windows 10 version.
- **RelatedCV** The previous correlation vector that was used byfollowing fields are available: autoStartRunCount The auto start run count of Update Assistant. clientID The ID of the client before swapping with a new onecurrent release of Update Assistant.
- **RepeatFailCount****launchMode** Indicates whetherthe type of launch performed. launchTypeReason A bitmask of all the reasons for type of launch. triggerTaskSource Indicates which task launches this specific pieceinstance. UALaunchRunCount Total number of contenttimes Update Assistant launched. Microsoft.Windows.UpdateAssistant.Orchestrator.RestoreEventId The event sends basic info on whether the Windows 10 update notification has previously failedlaunched.
- **RevisionNumber** IdentifiesThe following fields are available: clientID ID of the revision numbercurrent release of this specificpiece of contentUpdate Assistant.
- **ServiceGuid** Identifierrestore Reason All the reasons for the service torestore. triggerTaskSource Indicates whichtask launches this instance. Update events Update360Telemetry.UpdateAgent_DownloadRequest This event sends data during the softwaredistribution client is connecting (download request phase of updating Windows. The following fields are available: DeletedCorruptFiles Indicates if UpdateAgent found any corrupt payload files and whether the payload was deleted. ErrorCode The error code returned for the current download request phase. FlightId Unique ID for each flight. ObjectId Unique value for each Update, Microsoft Store, etc.).
- **StatusCode****Agent mode. PackageCountOptional Number of optional packages requested. PackageCountRequired Number of required packages requested. PackageCountTotal Total number of packages needed. PackageCountTotalCanonical Total number of canonical packages. PackageCountTotalDiff Total number of diff packages. PackageCountTotalExpress Total number of express packages. PackageSizeCanonical Size of canonical packages in bytes PackageSizeDiff Size of diff packages in bytes PackageSizeExpress Size of express packages in bytes RangeRequestState Represents the state of the download range request. RelatedCV Correlation vector value generated from the latest USO scan.** Result codeResult of the event (successdownload request phase of update. ScenarioId The scenario ID. Example: MobileUpdate, cancellationDesktopLanguagePack, failure code HResult).
- **TargetGroupId** For drivers targeted to a specific device modelDesktopFeatureOnDemand, thisor DesktopDriverUpdate SessionId Unique value for each Update Agent mode attempt. UpdateId Unique ID indicatesfor each update. Update360Telemetry.Updat

eAgent_Initialize This event sends data during the distribution groupinitialize phase of devicesreceiving that driverupdating Windows.

- **TargetingVersion** For drivers targeted to a specific device model, this isThe following fields are available: ErrorCode The error code returned for the version number ofcurrent initialize phase. FlightId Unique ID for each flight. FlightMetadata Contains the driversFlightId and the build being distributed toflighted. ObjectId Unique value for each Update Agent mode. RelatedCV Correlation vector value generated from the devicelatest USO scan.Result Result of the initialize phase of update. 0 = Succeeded, 1 = Failed, 2 = Cancelled, 3 = Blocked, 4 = BlockCancelled ScenarioId The scenario ID. Example: MobileUpdate, DesktopLanguagePack, DesktopFeatureOnDemand, or DesktopDriverUpdate SessionData Contains instructions to update agent for processing FODs and DUICs (Null for other scenarios). SessionId Unique value for each Update Agent mode attempt .
- **UpdateId** The identifier associated withUnique ID for each update. Update360Telemetry.UpdateAgent_Install This event sends data during the specific pieceinstall phase of contentupdating Windows.
- **UpdateImportance** IndicatesThe following fields are available: ErrorCode The error code returned for the importancecurrent install phase. FlightId Unique ID for each flight. ObjectId Unique value for each Update Agent mode. RelatedCV Correlation vector value generated from the latest scan. Result Result of a driverthe install phase of update. 0 = Succeeded 1 = Failed, andwhy it received that importance level (0-Unknown2 = Cancelled, 1-Optional3 = Blocked, 2-Important-DNF4 = BlockCancelled ScenarioId The scenario ID. Example: MobileUpdate, 3-Important-GenericDesktopLanguagePack, 4-Important-OtherDesktopFeatureOnDemand, 5-Recommended).
- **UsedSystemVolumeor DesktopDriverUpdate SessionId Unique value for each Update Agent mode attempt. 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: FlightId Unique ID for each flight. Mode**Indicates whetherthat the device's main system storage driveUpdate Agent mode that has started. 1 = Initialize, 2 = DownloadRequest, 3 = Install, 4 = Commit ObjectId Unique value for each Update Agent mode. RelatedCV The correlation vector value generated from the latest scan. ScenarioId The scenario ID. Example: MobileUpdate, DesktopLanguagePack, DesktopFeatureOnDemand, or an alternate storage drive was usedDesktopDriverUpdate SessionId Unique value for each Update Agent mode attempt.
- **WUDeviceIDUpdateId** Unique device ID controlled by the software distribution clientfor each update.

SoftwareUpdateClientTelemetryUpdate360Telemetry.TaskRunUpdateAgent_SetupBoxLaunch

StartThis eventsends data during the launching of the setup box when updating Windows. The following fields are available: FlightId Unique ID for Server Initiated Healing clienteach flight. See EventScenario fieldObjectId Unique value for specifics (each Update Agent mode. Quiet Indicates whether setup is running in quiet mode. 0 = false 1 = true RelatedCV Correlation vector value generated from the latest scan. SandboxSize The size of the sandbox folder on the device. ScenarioId The scenario ID. Example: MobileUpdate, DesktopLanguagePack, DesktopFeatureOnDemand, or DesktopDriverUpdate SessionId Unique value for exampleeach Update Agent mode attempt. SetupMode Setup mode 1 = predownload,2 = install, 3 = finalize UpdateId Unique ID for each update. Upgrade events Setup360Telemetry.Downlevel This event sends data indicating that the device hasstarted/completed).the downlevel phase of the upgrade, to help keep Windows up-to-date and secure.

The following fields are available:

- **CallerApplicationName** Name of application makingClientId If using Windows Update, this will be the Windows Update requestclient ID that is passed to Setup. UsedIn Media setup, the default value is Media360, but it can be overwritten by the caller to identify context of requesta unique value.
- **ClientVersion** VersionHostOSBuildNumber The build number of the software distribution clientdownlevel OS.
- **CmdLineArgs** Command line argumentsHostOsSkuName The operating system edition which is running Setup360 instance (downlevel OS). InstanceId A unique GUID that identifies each instance of setuphost.exe. ReportId In the Windows Update scenario, this is the updateID that is passed in byto Setup. In media setup, this is the caller.
- **EventInstanceID** A globally unique identifierGUID for the event instanceinstall.
- **EventScenario** Indicateswim. Setup360Extended More detailed information about phase/action when the purposepotential failure occurred. Setup360Mode The phase of the eventSetup360 (scan startedfor example,Predownload, Install, Finalize, Rollback). Setup360Result The result of Setup360 (HRESULT used to diagnose errors). Setup360Scenario The Setup360 flow type (for example, Boot, Media, Update, MCT). SetupVersionBuildNumber The build number of Setup360 (build number of the target OS). State Exit state of given Setup360 run. Example: succeeded, failed, etc.).
- **ServiceGuid** Identifier forblocked, cancelled. TestId An ID that uniquely identifies a group of events. Wuld This is the serviceWindows Update Client ID. In the Windows Update scenario, this is the same as the clientId. Setup360Telemetry.Finalize This event sends data indicating that the device has started the phase of finalizing the upgrade, to whichhelp keep Windows up-to-date and secure. The following fields are available: ClientId With Windows Update, this will be the softwaredistributionWindows Update clientID that is connectingpassed to Setup. In Media setup, default value is Media360, but can be overwritten by the calle

r to a unique value. HostOSBuildNumber The build number of the previous OS. HostOsSkuName The OS edition which is running Setup360 instance (previous OS). InstanceId A unique GUID that identifies each instance of setuphost.exe ReportId With Windows Update, Microsoft Storethis is the updateID that is passed to Setup. In media setup, etc.).

- **StatusCodethis is the GUID for the install.wim. Setup360Extended More detailed information about the phase/action when the potential failure occurred. Setup360Mode The phase of Setup360. Example: Predownload, Install, Finalize, Rollback. Setup360** ResultThe result of Setup360. This is an HRESULT error codethat is used to diagnose errors. Setup360Scenario The Setup360 flow type. Example: Boot, Media, Update, MCT. SetupVersionBuildNumber The build number of the eventSetup360(successbuild number of target OS). State The exit state of a Setup360 run. Example: succeeded, cancellationfailed, failure codeHResult).
- **WUDeviceID** Unique deviceblocked, cancelled. TestId ID controlled bythat uniquely identifies a group of events. Wuld This isthe software distribution clientWindows Update Client ID.With Windows Update, this is the same as the clientId.

SoftwareUpdateClientTelemetrySetup360Telemetry.UninstallOsUninstall

UninstallThis event for target update onsends data regarding OS updates and upgrades from Windows Update Client7, Windows 8, and Windows 10. See EventScenario field for specifics (for exampleSpecifically, Started/Failed/Succeeded).it indicates the outcome of an OS uninstall.

The following fields are available:

- **BundleIdClientId 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. HostOSBuildNumber** The identifierassociated withbuild number of the specific content bundleprevious OS. This should notHostOsSkuName The OS edition which is running the Setup360 instance (previous OS). InstanceId A unique GUID that identifies each instance of setuphost.exe. ReportId For Windows Update, this will be all zeros if the bundleID was foundWindows Update client ID that is passed to Setup.
- **BundleRepeatFailCount** Indicates whetherIn Media setup, this particular update bundle previously failedis the GUID for the install.
- **BundleRevisionNumber** Identifieswim. Setup360Extended Detailed information about the revisionphase or action when the potential failure occurred. 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. Setup360Scenario The Setup360 flow type. Example: Boot, Media, Update, MCT SetupVersionBuildNumber The build number of the contentbundleSetup360 (build number of target OS). State Exit state of a Setup360 run.
- **CallerApplicationName** NameExample: succeeded, failed, blocked, cancelled. TestId ID that uniquely identifies a groupofevents. Wuld Windows Update client ID. Setup360Telemetry.PostRebootInstall This event sends data indicating that the application making device has invoked thepost reboot install phase of the upgrade, to help keep Windowsup-to-date. The following fields are available: ClientId With Windows Update request, this is the Windows Update client ID that is passed to Setup. UsedIn Media setup, the default value is Media360, but can be overwritten by the caller to identify context of request a unique value.
- **ClientVersion** VersionHostOSBuildNumber The build number of the software distribution clientprevious OS.
- **CommonPropsHostOsSkuName The OS edition which is running Setup360 instance (previous OS). InstanceId** A bitmask forfuture flags associated with theunique GUID that identifies each instance of setuphost.exe. ReportId With Windows Update client behavior. There, this is no value being reported in this field right nowthe updateID that is passed to Setup. Expected value forIn media setup, this field is 0the GUID for the install.
- **DriverPingBack** Containswim. Setup360Extended Extension of result - more granular information aboutphase/action when the previous driver and system statepotential failure happened Setup360Mode The phase of Setup360.
- **DriverRecoveryIdsExample: Predownload, Install, Finalize, Rollback Setup360Result** The listresult of identifiers that couldbe Setup360. This is an HRESULT error code that's used for uninstalling the drivers when a recovery is requiredto diagnose errors.
- **EventInstanceID** A globally unique identifier for event instanceSetup360Scenario The Setup360 flow type.
- **EventScenario** Indicates the purposeExample: Boot, Media, Update, MCT SetupVersionBuildNumber The build number of theeventSetup360 (build number of target OS). State The exit state of a scan startedSetup360 run. Example: succeeded, succeded,failed, etc.).
- **EventType** Indicatesblocked, cancelled TestId A string to uniquely identify a group of events. Wuld This is theWindows Update Client ID. With Windows Update, this is the same as ClientId. Setup360Telemetry.PreDownloadQuiet This event typesends data indicating that the device has invoked the predownload quiet phase of the upgrade, to help keep Windows up to date. Possible valuesThe following fields are "Child", "Bundle", "Release" or "Driver".
- **ExtendedStatusCode** Secondary status code for certain scenarios where StatusCodeavailable: ClientId Using Windows Update, this will be the Windows Update client ID that is not specific enoughpassed to Setup.

- **FeatureUpdatePause** Indicates whether feature OS updates are paused on~~In Media setup, default value is Media360, but can be overwritten by~~ the device~~caller to a unique value~~.
- **FlightBuildNumber** Indicates the~~HostOSBuildNumber The~~ build number of the flight~~previous OS~~.
- **FlightId~~HostOsSkuName~~** The specific ID~~OS edition which is running Setup360 instance (previous operating system). InstanceId A unique GUID that identifies each instance~~ of~~setuphost.exe. ReportId Using Windows Update, this is~~ the flight thedevice~~updateI D that~~ is getting~~passed to Setup~~.
- **HandlerType** Indicates~~In media setup, this is~~ the kind~~GUID for the install.wim. Setup360Extended Detailed information about the phase/action when the potential failure occurred. Setup360Mode The phase~~ of content (app~~Setup360. Example: Predownload~~, driver~~Install~~, windows patch~~Finalize~~, etc.).
- **HardwareId** If the download was for a driver targeted~~Rollback. Setup360Result The result of Setup360. This is an HRESULT error code that is used~~ to~~diagnose errors. Setup360Scenario The Setup360 flow type. Example: Boot, Media, Update, MCT. SetupVersionBuildNumber The build number of Setup360 (build number of target OS). State The exit state of~~ a particulardevice model~~Setup360 run. Example: succeeded~~, this~~failed, blocked, canceled. TestId~~ ID indicates the model~~that uniquely identifies a group~~ of~~events. WuId This is~~ the device~~Windows Update Client ID~~.
- **IsFinalOutcomeEvent** Indicates whether~~Using Windows Update, this is the same as the clientId. Setup360Telemetry.PreDownloadUX This~~ event signals~~sends data regarding OS Updates and Upgrades from Windows 7.X, Windows 8.X, Windows 10 and RS, to help keep Windows up-to-date and secure. Specifically, it indicates~~ the end~~outcome~~ of the~~PredownloadUX portion of the~~ update/upgrade process.
- **IsFirmware** Indicates whether an update was a firmware update~~The following fields are available: ClientId For Windows Update, this will be the Windows Update client ID that is passed to Setup~~.
- **IsSuccessFailurePostReboot** Indicates whether an initial success was then~~In Media setup, default value is Media360, but can be overwritten by the caller to~~ a failure after a reboot~~unique value~~.
- **IsWUfBDualScanEnabled** Flag indicating whether WU-for-Business dual scan~~HostOSBuildNumber The build number of the previous operating system. HostOsSkuName The OS edition which~~ is enabled on~~running~~ the device~~Setup360 instance (previous operating system). InstanceId Unique GUID that identifies each instance of setuphost~~.
- **IsWUfBEnabled** Flag indicating whether WU-for-Business~~exe. ReportId For Windows Update, this will be the Windows Update client ID that~~ is enabled on~~passed to Setup. In Media setup, this is~~ the device.
- **MergedUpdate** Indicates whether an OS update and a BSP update were merged~~GUID~~ for~~the~~ install.
- **ProcessName** Process name~~wim. Setup360Extended Detailed information about the phase/action when the potential failure occurred. 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. Setup360Scenario The Setup360 flow type. Example: Boot, Media, Update, MCT. SetupVersionBuildNumber The build number of Setup360 (build number of~~ the callerwho initiated API calls into~~target OS). State The exit state of~~ the software distribution~~Setup360 run. Example: succeeded, failed, blocked, cancelled. TestId ID that uniquely identifies a group of events. WuId Windows Update~~ client~~ID~~.
- **QualityUpdatePause** Indicates whether quality OS updates are paused on~~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~~.
- **RelatedCV** The previous correlation vector~~following fields are available: ClientId With Windows Update, this will be the Windows Update client ID~~ that was used~~is passed to Setup. In Media setup, default value is Media360, but can be overwritten~~ by the client before swapping with~~caller to~~ a new one~~unique value~~.
- **RepeatFailCount** Indicates whether this specific piece~~HostOSBuildNumber The build number~~ of content previously failed~~the previous OS~~.
- **RevisionNumber** Identifies the revision number~~HostOsSkuName The OS edition which is running Setup360 instance (previous OS). InstanceId A unique GUID that identifies each instance~~ of~~setuphost.exe ReportId With Windows Update,~~ this specificpiece of content~~is the updateID that is passed to Setup~~.
- **ServiceGuid** Identifier~~In media setup, this is the GUID~~ for the service to which~~install.wim. Setup360Extended Detailed information about~~ the software distribution client~~phase/action when the potential failure occurred. Setup360Mode The phase of Setup360. Example: Predownload, Install, Finalize, Rollback. Setup360Result The result of Setup360. This~~ is connecting~~an HRESULT error code that can be used to diagnose errors. Setup360Scenario Setup360 flow type~~ (Windows~~Boot, Media,~~Update, Microsoft Store~~MCT). SetupVersionBuildNumber The build number of Setup360 (build number of target OS). State The exit state of a Setup360 run. Example: succeeded~~, etc.).
- **StatusCode** Result code~~failed, blocked, cancelled. TestId A string to uniquely identify a group~~ of~~events. WuId This is~~the~~Windows Update Client ID. With Windows Update, this is the same as the clientId. Setup360Telemetry.PreInstallUX This~~event (success~~sends data regarding OS updates and upgrades from Windows 7~~, cancellation~~Windows 8~~, failure code HResult).
- **TargetGroupId** For drivers targeted~~and Windows 10,~~ to a specific device model~~help keep Windows up-to-date. Specifically~~, this~~ID~~it~~ indicates the distribution group~~outcome~~ of devices receiving that driver~~the PreinstallUX portion of the update process~~.

- **TargetingVersion** **The following fields are available: ClientId** For drivers targeted to a specific device model 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 version caller to a unique value. HostOSBuildNumber The build number of the drivers being distributed previous OS. HostOsSkuName The OS edition which is running the Setup360 instance (previous OS). 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 device GUID for the install.
- **UpdateId** Identifier associated with wim. Setup360Extended Detailed information about the specific piece phase/action when the potential failure occurred. Setup360Mode The phase of content Setup360.
- **UpdateImportance** Indicates the importance Example: Predownload, Install, Finalize, Rollback. Setup360Result The result of adriver and why it received Setup360. This is an HRESULT error code that importance level is used to diagnose errors. Setup360Scenario The Setup360 flow type, Example: Boot, Media, Update, MCT. SetupVersionBuildNumber The build number of Setup360 (0-Unknown build number of target OS). State The exit state of a Setup360 run. Example: succeeded, 1-Optional failed, 2-Important-DNF blocked, 3-Important-Generic cancelled. TestId A string to uniquely identify a group of events. WuId Windows Update client ID. Setup360Telemetry.Setup360 This event sends data about OS deployment scenarios, 4-Important-Other to help keep Windows up-to-date. The following fields are available: ClientId Retrieves the upgrade ID. In the Windows Update scenario, 5-Recommended).
- **UsedSystemVolume** Indicates whether this will be the device's main system storage drive or an alternate storage drive wasused Windows Update client ID.
- **WUDeviceID** Unique device ID controlled In Media setup, default value is Media360, but can be overwritten by the softwaredistribution client caller to a unique value. FieldName Retrieves the data point. FlightData Specifies a unique identifier for each group of Windows Insider builds. InstanceId Retrieves a unique identifier for each instance of a setup session. ReportId Retrieves the report ID. ScenarioId Retrieves the deployment scenario. Value Retrieves the value associated with the corresponding FieldName.

SoftwareUpdateClientTelemetry Setup360Telemetry.UpdateDetected Setup360DynamicUpdate

This event helps determine whether the device received supplemental content during an operating system upgrade, to help keep Windows up-to-date. Setup360Telemetry.UnexpectedEvent This event sends data about an AppX app indicating that the device has been updated from invoked the Microsoft Store unexpected event phase of the upgrade, including what app needs an update andwhat version/architecture is required, in order to understand and address problems with apps getting required updates help keep Windows up to date.

The following fields are available:

- **ApplicableUpdateInfo** Metadata for ClientId With Windows Update, this will be the updates which were detected asapplicable Windows Update client ID that is passed to Setup.
- **CallerApplicationName** The name provided In Media setup, default value is Media360, but can be overwritten by the caller whoinitiated API calls into the software distribution client to a unique value.
- **IntentPFNs** Intended application-set metadata for atomic update scenarios.
- **NumberOfApplicableUpdates** **HostOSBuildNumber** The build number of updates ultimately deemed applicable to the systemafter the detection process is complete previous OS.
- **RelatedCV** **HostOsSkuName** The OS edition which is running Setup360 instance (previous Correlation Vector OS). InstanceId A unique GUID that was used before swapping with a new one identifies each instance of setuphost.
- **ServiceGuid** An ID exe ReportId With Windows Update, this is the updateID that represents which service is passed to Setup. In media setup, this is the software distribution client GUID for the install.wim. Setup360Extended Detailed information about the phase/action when the potential failure occurred. Setup360Mode The phase of Setup360. Example: Predownload, Install, Finalize, Rollback. Setup360Result The result of Setup360. This is connecting an HRESULT error code that can be used to diagnose errors. Setup360Scenario The Setup360 flow type. Example: Boot, Media, Update, MCT. SetupVersionBuildNumber The build number of Setup360 (build number of target OS). State The exit state of a Setup360 run. Example: succeeded, failed, blocked, cancelled. TestId A string to uniquely identify a group of events. WuId This is the Windows Update Client ID. With Windows Update, this is the same as the clientId. Windows Error Reporting events Microsoft Store. Windows.WERVertical.OSCrash This event sends binary data from the collected dump file whenever a bug check occurs, etc.).
- **WUDeviceID** **to help keep Windows up to date.** The unique is the OneCore version of this event. The following fields are available: BootId Uint32 identifying the boot number for this device ID controlled by. BugCheckCode Uint64 "bugcheck code" that identifies a proximate cause of the software distribution client bug check. BugCheckParameter1 Uint64 parameter providing additional information. BugCheckParameter2 Uint64 parameter providing additional information. BugCheckParameter3 Uint64 parameter

providing additional information. BugCheckParameter4 Uint64 parameter providing additional information. DumpFileAttributes Codes that identify the type of data contained in the dump file DumpFileSize Size of the dump file IsValidDumpFile True if the dump file is valid for the debugger, false otherwise ReportId WER Report Id associated with this bug check (used for finding the corresponding report archive in Watson).

System Resource Usage MonitorWindows Store events

Microsoft.Windows.SrumStore.SdpPartner.CpuUsageReportApplication

Report application event for Windows Store client. Microsoft.Windows.StoreAgent.Telemetry.AbortedInstallation This event providesinformation on CPU usageis 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:

- **UsageMaxAggregatedPackageFullNames** The maximumnames of hourly average CPU usageall packages to be downloaded and installed.
- **UsageMeanAttemptNumber Number of retry attempts before it was canceled. BundleId** The meanItem Bundle ID. CategoryId The Item Category ID. ClientAppId The identity of hourly average CPU usagethe app that initiated this operation.
- **UsageMedianHResult** The medianresult code of hourly average CPU usagethe last action performed before this operation.
- **UsageTwoHourMaxMeanIsBundle Is this a bundle? IsInteractive Was this requested by a user? IsMandatory Was this a mandatory update? IsRemediation Was this a remediation install? IsRestore Is this automatically restoring a previously acquired product? IsUpdate Flag indicating if this is an update. ParentBundleId** The meanproduct ID of the maximumparent (if this product is part of every two houra bundle). PFN The product family name of hourly average CPU usagethe product being installed.
- **UsageTwoHourMedianMeanProductId** The meanidentity of the medianpackage or packages being installed. SystemAttemptNumber The total number of every two hourautomatic attempts at installation before it was canceled. UserAttemptNumber The total number of hourly average CPU usageuser attempts at installation before it was canceled.WUContentId Licensing identity of this package.

Microsoft.Windows.SrumStoreAgent.SdpTelemetry.NetworkUsageBeginGetInstalledContentIds

This event provides information on network usageis 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:

- **AdapterGuidAggregatedPackageFullNames** The unique IDnames of the adapterall package or packages to be downloaded and installed.
- **BytesTotalMaxAttemptNumber Total number of installation attempts. BundleId** The maximumidentity of the hourly averagebytes totalWindows Insider build that is associated with this product.
- **BytesTotalMeanCategoryId** The meanidentity of the hourly average bytes totalpackage or packages being installed.
- **BytesTotalMedianClientAppId** The medianidentity of the hourly average bytes totalapp that initiated this operation.
- **BytesTotalTwoHourMaxMeanIsBundle Is this a bundle? IsInteractive Was this requested by a user? IsMandatory Is this a mandatory update? IsRemediation Is this repairing a previous installation? IsRestore Is this an automatic restore of a previously acquired product? IsUpdate Is this a product update? ParentBundleId** The meanproduct ID of the maximumparent (if this product is part of every two hoursa bundle). PFN The name of hourly average bytes totalall packages to be downloaded and installed.
- **BytesTotalTwoHourMedianMeanPreviousHResult** The meanprevious HResult code. PreviousInstallState Previous installation state before it was canceled. ProductId The name of the medianpackage or packages requested for installation. RelatedCV Correlation Vector of every two houra previous performed

action on this product. SystemAttemptNumber Total number of hourly average bytes totalautomatic attempts to install before it was canceled.

- **LinkSpeedUserAttemptNumber Total number of user attempts to install before it was canceled. WUContentId** The adapterlink speedWindows Update content ID.

Upgrade events

FacilitatorTelemetryMicrosoft.DCATDownloadWindows.StoreAgent.Telemetry.CompleteInstallOperationRequest

This event indicates whether devices received additionalis sent at the end of app installations or critical supplemental contentduring an OS Upgrade,updates to help keep Windows up-to-date and secure.

The following fields are available:

- **DownloadSize** Download sizeCatalogId The Store Product ID of payloadthe app being installed.
- **ElapsedTime** Time taken to download payloadHResult HResult code of the action being performed.
- **MediaFallbackUsed** Used to determine if we used Media CompDBs to figure outIsBundle Is this a bundle? PackageFamilyName The name of the package requirements for the upgradebeing installed.
- **ResultCode** Result returned byProductId The Store Product ID of the Facilitator DCAT callproduct being installed.
- **Scenario** Dynamic update scenario (Image DU, or Setup DU).
- **Type** TypeSkuId Specific edition of package that was downloadedthe item being installed.

FacilitatorTelemetryMicrosoft.InitializeDUWindows.StoreAgent.Telemetry.EndAcquireLicense

This event determines whether devices received additional or critical supplemental content during an OS upgradeis 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:

- **DCATUrlAggregatedPackageFullNames Includes a set of package full names for each app that is part of an atomic set. AttemptNumber** The Delivery Catalog (DCAT) URL we send the requesttotal number of attempts toacquire this product.
- **DownloadRequestAttributesCategoryId** The attributes we send to DCATidentity of the package or packages being installed.
- **ResultCodeClientAppId** Theidentity of the app that initiated this operation. HResult HResult code to show the result returnedfrom the initiation of Facilitator with the URLoperation (success/attributes.
- **Scenario** Dynamic Update scenario (Image DU, or Setup DUfailure).
- **UrlIsBundle Is this a bundle? IsInteractive Did the user initiate the installation? IsMandatory Is this a mandatory update? IsRemediation Is this repairing a previous installation? IsRestore Is this happening after a device restore? IsUpdate Is this an update? ParentBundledId** The Delivery Catalogproduct's parent bundle ID. PFN Product Family Name of the product being installed. ProductId The Store Product ID for the product being installed. SystemAttemptNumber The number of attempts by the system to acquire this product. UpdateId The update ID (DCATif this is an update) URL we sendUserAttemptNumber The number of attempts by the requestuser toacquire this product WUContentId The Windows Update content ID.
- **Version** Version of Facilitator.

Setup360Microsoft.Windows.StoreAgent.Telemetry.Setup360DynamicUpdateEndDownload

This event helps determine whether the device received supplemental content duringis sent after an operating system upgrade,app is downloaded to help keep Windows up-to-dateand secure.

The following fields are available:

- **FlightData** Specifies a unique identifier for each groupAggregatedPackageFullNames The name ofall packages to be downloaded and installed. AttemptNumber Number of retry attempts before it was canceled. BundleId The identity of theWindows Insider buildsbuild associated with this product.

- **InstanceId** Retrieves a unique identifier for each instanceCategoryId The identity of a setup sessionthe package or packages being installed.
- **Operation** Facilitator's last knownClientAppId The identity of the app that initiated this operation (scan, DownloadSize The total size of the download, etc.).
- **ReportId** ID for tying together events stream side.
- **ResultCode** Result returnedExtendedHResult Any extended HResult error codes. HResult The result code of the last action performed. IsBundle Is this a bundle? IsInteractive Is this initiated by Setup for the entire operation.
- **Scenario** Dynamic Update scenariouser? IsMandatory Is this a mandatory installation? IsRemediation Is this repairing a previous installation? IsRestore Is this a restore of a previously acquired product? IsUpdate Is this an update? ParentBundleId The parent bundle ID (Image DU, or Setup DUif it's part of a bundle).
- **ScenarioId** IdentifiesPFN The Product Family Name of the update scenarioapp being download.
- **TargetBranch** BranchProductId The Store Product ID for the product being installed. SystemAttemptNumber The numberofattempts by the target OSsystem to download.
- **TargetBuild** BuildUserAttemptNumber The number ofattempts by the target OSuser to download.WUContentId The Windows Update content ID.

~~Microsoft.~~Windows as a Service diagnosticevents.~~StoreAgent.Telemetry.EndFrameworkUpdate This event is sent when an app update requires an updated Framework package and the process starts to download it. It is 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.WaaSMedic~~StoreAgent.~~SummaryEvent~~Telemetry.EndGetInstalledContentIds~~

ResultThis event is sent after sending the inventory of the WaaSMedicproducts 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 to help keep Windows up-to-date and secure.

The following fields are available:

- **callerApplication~~AggregatedPackageFullNames~~** The name~~names~~ ofall packages to be downloaded and installed. AttemptNumber The number of retry attempts before it was canceled. BundleId The identity of the calling applicationbuild associated with this product.
- **detectionSummary** ResultCategoryId The identity of each applicable detectionthe package or packages being installed. ClientAppId The identity of the app that was runinitiated this operation.
- **featureAssessmentImpact** WaaS Assessment impact for feature updates~~ExtendedHResult The extended HResult error code~~.
- **hrEngineResult** Error~~HResult The result~~ code from~~of~~ the engine operation~~last action performed~~.
- **isInteractiveMode~~IsBundle Is this a bundle? IsInteractive Is this an interactive installation? IsMandatory Is this a mandatory installation? IsRemediation Is this repairing a previous installation? IsRestore Is this automatically restoring a previously acquired product? IsUpdate Is this an update? ParentBundleId~~** The user started~~product ID of the parent (if this product is part of~~ a run~~bundle). PFN Product Family Name~~ of WaaSMedic~~the product being installed~~.
- **isManaged** Device~~ProductId The Store Product ID for the product being installed. SystemAttemptNumber The total number of system attempts. UserAttemptNumber The total number of user attempts. WUContentId The Windows Update content ID. Microsoft.Windows.StoreAgent.Telemetry.EndScanForUpdates This event~~ is managed~~sent after a scan~~ for~~product~~ updates~~to determine if there are packages to install~~.
- **isWUConnected** Device is connected~~It's used to~~help keep Windows Update~~up-to-date and secure~~.
- **noMoreActions** No more~~The following fields are available: ClientAppId The identity of the app that initiated this operation. HResult The result code of the last action performed. IsApplicability Is this request to only check if there are any~~ applicable diagnostic spackages to install? IsInteractive Is this user requested? IsOnline Is the request doing an online check? Microsoft~~.
- **qualityAssessmentImpact** WaaS Assessment impact~~Windows.StoreAgent.Telemetry.EndSearchUpdatePackages This event is sent after searching~~ for quality updates~~update packages to install~~.
- **remediationSummary** Result~~It is used to help keep Windows up-to-date and secure. The following fields are available: AggregatedPackageFullNames The names~~ of each~~all packages to be downloaded and installed. AttemptNumber The total number of retry attempts before it was canceled. BundleId The identity of the build associated with this product. CategoryId The identity of the package or packages being installed. ClientAppId The identity of the~~

app that initiated this operation. HResult The result code of the last action performed on. IsBundle Is this a device to fixbundle? Is Interactive Is this user requested? IsMandatory Is this a mandatory update? IsRemediation Is this repairing a previous installation ? IsRestore Is this restoring previously acquired content? IsUpdate Is this an invalid stateupdate? ParentBundleId The product ID of the parent (if this product is part of a bundle). PFN The name of the package or configuration that's preventingpackages requ ested for install. ProductId The Store Product ID for the device from getting updatesproduct being installed. For example,ifSyste mAttemptNumber The total number of system attempts. UserAttemptNumber The total number of user attempts. WUContentId The Windows
Update servicecontent ID. Microsoft.Windows.StoreAgent.Telemetry.EndStageUserData This eventis turned off, the fixsent after r estoring user data (if any) that needs to be restored following a product install. It isused to turnthe it back onkeep Windows up-to-date and secure.

- **usingBackupFeatureAssessment** Relying on backup feature assessmentThe following fields are available: AggregatedPackageF ullNames The name of all packages to be downloaded and installed.
- **usingBackupQualityAssessment** Relying on backup quality assessmentAttemptNumber The total number of retry attempts bef ore it was canceled.
- **usingCachedFeatureAssessment** WaaS Medic run did not get OSBundleId The identity of the build age fromassociated with this product. CategoryId The identity of the network on the previous runpackage or packages being installed.
- **usingCachedQualityAssessment** WaaS Medic run did not get OS revision age fromClientAppId The identity of the network onth e previous runapp that initiated this operation.
- **versionString** VersionHResult The result code of the WaaSMedic enginelast action performed.
- **waasMedicRunMode** Indicates whetherIsBundle Is this was a background regular runbundle? IsInteractive Is this user requested ? IsMandatory Is this a mandatory update? IsRemediation Is this repairing a previous installation? IsRestore Is this restoring previ ously acquired content? IsUpdate Is this an update? ParentBundleId The product ID of
the medicparent (if this product is part of a bundle). PFN The name of the package or whether it was triggered by a user launchin gpackages requested for install. ProductId The Store Product ID for the product being installed. SystemAttemptNumber The total number of system attempts. UserAttemptNumber The total number of system attempts. WUContentId The Windows
Update Troubleshootercontent ID.

Microsoft.Windows Updateevents.StoreAgent.Telemetry.EndUpdateMetadataPrepare This even t 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.UpdateStoreAgent.DeviceUpdateAgentT elemetry.UpdateAgentAnalysisSummaryFulfillmentComplete

This
event collects information regardingis sent at the stateend of devices and drivers on the system following a reboot after thean app insta ll phase of the new device manifest UUP (Unified Update Platform)or update scenario which is used to install a devicemanifest describin g a set of driver packageshelp keep Windows up-to-date and secure.

The following fields are available:

- **activated** WhetherFailedRetry Indicates whether the entire device manifestinstallation or update is considered activated and inus eretry was successful.
- **analysisErrorCount** How many driver packages that could not be analyzed because errors were hit duringHResult The HResult c ode of the analysisoperation.
- **flightId** UniquePFN The Package Family Name of the app that is being installed or updated. ProductId The product ID for eachfli ghtof the app that is being updated or installed.
- **missingDriverCount** How many driver packages that were deliveredbyMicrosoft.Windows.StoreAgent.Telemetry.FulfillmentInitia te This event is sent at the device manifest thatbeginning of an app install or update to help keep Windows up-to-date and secure. The following fields are missing fromavailable: PFN The Package Family Name of the systemapp that is being in stalled or updated.
- **missingUpdateCount** How many updates that were partProductId The product ID of
the device manifestapp thatis being updated or installed. Microsoft.Windows.StoreAgent.Telemetry.InstallOperationRequest This event is sent when a product install or update is initiated, to help keep Windows up-to-
date and secure. The following fields are missingavailable: BundleId The identity of the build associated with this product. Catalo gId If this product is froma private catalog, the systemStore Product ID for the product being installed.
- **objectId** Unique valueProductId The Store Product ID for each diagnostics sessionthe product being installed.

- **publishedCount** How many drivers packages that were deliveredSkuId Specific edition ID 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 bya user or the device manifest thatsystem), to help keep Windows up-to-date and secure. The following fields are published and available: AggregatedPackageFullNames The names of all packages to be used ondevicesdownloaded and installed.
- **relatedCV** Correlation vector value generated fromAttemptNumber The total number of retry attempts before it was canceled. BundleId The identity of the latest USO scanbuild associated with this product.
- **scenarioId** IndicatesCategoryId The identity of the update scenariopackage or packages being installed.
- **sessionId** Unique value for each update session.
- **summary** A summary string that contains some basic information about driver packages that are partClientAppId The identityof the device manifest and any devices on the systemapp that those driver packages match oninitiated this operation.
- **summaryAppendError** A Boolean indicating if there wasIsBundle Is this a bundle? IsInteractive Is this user requested? IsMandatory Is this a mandatory update? IsRemediation Is this repairing a previous installation? IsRestore Is this restoring previously acquired content? IsUpdate Is this an error appending more information toupdate? ParentBundleId The product ID of the summary stringparent (if this product is part of a bundle). PFN The Product Full Name.
- **truncatedDeviceCount** How many devices are missing fromPreviousHResult The result code of the summary string due tothere not being enough room inlast action performed before this operation. PreviousInstallState Previous state before the stringinstallation or update was paused.
- **truncatedDriverCount** How many driver packages are missing fromProductId The Store Product ID for the summary string duet o there notproduct being enough room in the stringinstalled.
- **unpublishedCount** How many drivers packages that were delivered by the device manifest that are still unpublished andunavailable to be usedRelatedCV Correlation Vector of a previous performed action on devicesthis product.
- **updateId** UniqueSystemAttemptNumber The total number of system attempts. UserAttemptNumber The total number of user attempts. WUContentId The Windows Update content ID for each Update.

Microsoft.Windows.UpdateStoreAgent.DeviceUpdateAgentTelemetry.UpdateAgentCommitResumeInstallation

This
event collects information regarding the final commit phase of the new device manifest UUP (Unified Update Platform)is sent when a product install or update scenario, which is usedresumed (either by a user or the system), to install a device manifestdescribing a set of driver packageshelp keep Windows up-to-date and secure.

The following fields are available:

- **errorCodeAggregatedPackageFullNames** The error code returned for the current session initializationnames of all packages to be downloaded and installed.
- **flightIdAttemptNumber** The unique identifier for each flightnumber of retry attempts before it was canceled.
- **objectIdBundleId** The unique GUID for each diagnostics sessionidentity of the build associated with this product.
- **relatedCV** A correlation vector value generated fromCategoryId The identity of the latest USO scanpackage or packages being installed.ClientAppId The identity of the app that initiated this operation. HResult The
- **result** Outcomecode of
the initializationlast action performed before this operation. IsBundle Is this a bundle? IsInteractive Is this user requested? IsMandatory Is this a mandatory update? IsRemediation Is this repairing a previous installation? IsRestore Is this restoring previously acquired content? IsUpdate Is this an update? IsUserRetry Did the user initiate the retry? ParentBundleId The product ID of
the sessionparent (if this product is part of a bundle). PFN The name of the package or packages requested for install.
- **scenarioId** IdentifiesPreviousHResult The previous HResult error code. PreviousInstallState Previous state before the Updatescenarioinstallation was paused.
- **sessionIdProductId** The unique valueStore Product ID for eachthe product being installed. RelatedCV Correlation Vector for the original install before it was resumed. SystemAttemptNumber The total number of system attempts. UserAttemptNumber The total number of user attempts. WUContentId The Windows Update content ID. Microsoft.Windows.StoreAgent.Telemetry.ResumeOperationRequest This event is sent when a product install or update sessionis resumed by a user or on installation retries, to help keep Windows up-to-date and secure.
- **updateId** The unique identifierfollowing fields are available: ProductId The Store Product ID for each Updatethe product being installed.

Microsoft.Windows.UpdateStoreAgent.DeviceUpdateAgentTelemetry.UpdateAgentDownloadRequestSearchForUpdateOperationRequest

This event collects information regarding~~is sent when searching for update packages to install, to help keep Windows up-to-date and secure. The following fields are available: CatalogId The Store Catalog ID for~~ the download request phase~~product being installed. ProductId The Store Product ID for the product being installed. SkuId Specfic edition~~ of the new device manifest UUP(Unified Update Platform)~~app being updated. Microsoft.Windows.StoreAgent.Telemetry.UpdateAppOperationRequest This event occurs when an~~ update scenario~~is requested for an app~~, which~~to help keep Windows up-to-date and secure. The following fields are available: PFamN The name of the app 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 install a device manifest describing a set of driver packages~~understand and address problems regarding downloads~~.

The following fields are available:

- **deletedCorruptFiles** Indicates if UpdateAgent found any corrupt payload files and whether~~background Is~~ the payload~~download being done in the background? bytesFromCDN The number of bytes received from a CDN source. bytesFromGroupPeers The number of bytes received from a peer in the same group. bytesFromIntPeers The number of bytes received from peers not in the same LAN or in the same group. bytesFromPeers The number of bytes received from a peer in the same LAN. cdnErrorCodes A list of CDN connection errors since the last FailureCDNCommunication event. cdnErrorCounts The number of times each error in cdnErrorCodes~~ was deleted~~encountered.clientTelId A random number used for device sampling. doErrorCode The Delivery Optimization error code that was returned.~~
- **errorCode** The error code~~that was~~ returned for~~. experimentId When running a test, this is used to correlate events that are part of~~ the current session initialization~~same test~~.
- **flightId**~~**fileID**~~ The unique identifier for each flight~~ID of the file being downloaded~~.
- **objectId** Unique value for each Update Agent mode~~isVpn Is the device connected to a Virtual Private Network? scenarioID The ID of the scenario~~.
- **packageCountOptional** Number~~sessionID The ID~~ of optional packages requested~~the file download session~~.
- **packageCountRequired** Number~~updateID The ID~~ of required packages requested~~the update being downloaded~~.
- **packageCountTotal** Total~~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: background Is the download a background download? bytesFromCDN The~~ number of packages needed~~bytes received from a CDN source~~.
- **packageCountTotalCanonical** Total~~bytesFromGroupPeers The~~ number of canonical packages~~bytes received from a peer in the same domain group~~.
- **packageCountTotalDiff** Total~~bytesFromIntPeers The~~ number of diff packages~~bytes received from peers not in the same LAN or in the same domain group~~.
- **packageCountTotalExpress** Total~~bytesFromPeers The~~ number of express packages~~bytes received from a peer in the same LAN~~.
- **packageSizeCanonical** Size~~bytesRequested The total number~~ of canonical packages in bytes~~requested for download~~.
- **packageSizeDiff** Size~~cdnConnectionCount The total number~~ of diff packages~~connections made to the CDN. cdnErrorCodes A list of CDN connection errors since the last FailureCDNCommunication event. cdnErrorCounts The number of times each error~~ in by tes~~cdnErrorCodes was encountered~~.
- **packageSizeExpress** Size~~cdnIp The IP address~~ of express packages~~the source CDN. clientTelId A random number used for device sampling. doErrorCode The Delivery Optimization error code that was returned. downlinkBps The maximum measured available download bandwidth (~~ in bytes~~per second). downlinkUsageBps The download speed (in bytes per second). downloadMode The download mode used for this file download session~~.
- **rangeRequestState** Represents~~experimentId When running a test, this is used to correlate with other events that are part of~~the state~~same test. fileID The ID~~ of the download range request~~file being downloaded~~.
- **relatedCV** Correlation vector value generated from~~fileSize The size of~~ the latest USO scan~~file being downloaded~~.
- **result** Result~~groupConnectionCount The total number~~ of~~connections made to peers in~~ the download request phase~~same group. internetConnectionCount The total number~~ of update~~connections made to peers not in the same LAN or the same group~~.
- **scenarioId**~~**isVpn Is the device connected to a Virtual Private Network? lanConnectionCount**~~ The scenario ID~~total number of connections made to peers in the same LAN~~. Example: MobileUpdate, DesktopLanguagePack, DesktopFeatureOnDemand, orDesktopDriverUpdate.
- **sessionId** Unique value~~numPeers The total number of peers used~~ for each Update Agent mode attempt~~this download~~.
- **updateId** Unique~~restrictedUpload Is the upload restricted? scenarioID The~~ ID for each~~of the scenario. sessionID The ID of the download session. totalTimeMs Duration of the download (in seconds). updateID The ID of the~~ update~~being downloaded.uplinkBps The maximum measured available upload bandwidth (in bytes per second). uplinkUsageBps The upload speed (in bytes per second). usedMemoryStream Did the download use memory streaming?~~

Microsoft.WindowsOSG.UpdateDU.DeviceUpdateAgentDeliveryOptClient.UpdateAgentInitializeDownloadPaused

This event sends data for initializingrepresents a new update session for the new device manifest UUP (Unified Update Platform)update scenario, which istemporary suspension of a download with Delivery Optimization. It's used to install a device manifestdescribing a set of driver packagesunderstand and address problems regarding downloads.

The following fields are available:

background Is the download a background download? clientTelId A random number used for device sampling.

- **errorCode** The error codethat was returned for. experimentId When running a test, this is used to correlate with other events that are part of the current session initializationsame test.
- **flightIdfileID** The unique identifierID of the file being paused. isVpn Is the device connected to a Virtual Private Network? reasonCode The reason for each flightpausing the download.
- **flightMetadata** ContainsscenarioID The ID of the FlightId andscenario. sessionID The ID of the builddownload session. updateID The ID of the update being flightedpaused.
- **objectId** Unique value for each Update Agent modeMicrosoft.
- **relatedCV** Correlation vector value generated fromOSG.DU.DeliveryOptClient.DownloadStarted This event sends data describing the latest USO scanstart of a new download to enable Delivery Optimization.
- **result** ResultIt's used to understand and address problems regarding downloads. The following fields are available: background Indicates whether the download is happening in the background. cdnUrl The URL of the initialize phasesource CDN. clientTelId A random number used for device sampling. costFlags A set offlags representing network cost. deviceProfile Identifies the updateusage or form factor (such as Desktop, Xbox, or VM). diceRoll Random number used for determining if a client will use peering.doClientVersion The version of the Delivery Optimization client. doErrorCode The Delivery Optimization error code that was returned. downloadMode The download mode used for this file download session (CdnOnly = 0, Lan = Succeeded1, 1Group = Failed2, 2Internet = Cancelled3, 3Simple = Blocked99, 4Bypass = BlockCancelled100). errorCode The error code that was returned.
- **scenarioIdexperimentId ID used to correlate client/services calls that are part of the same test during A/B testing. fileID** The scenario IDof the file being downloaded. Example: MobileUpdate, DesktopLanguagePack, DesktopFeatureOnDemand, orDesktopDriverUpdatefilePath The path to where the downloaded file will be written.
- **sessionData** Contains instructionsgroupID ID for the group. isVpn Indicates whether the device is connected to update agentfor processing FODs and DUICsa Virtual Private Network. jobID The ID of the Windows Update job. minDiskSizeGB The minimum disk size (Nullin GB) policy set for other scenarios).
- **sessionId** Unique valuethe device to allow peering with delivery optimization. minDiskSizePolicyEnforced Indicates whether there is an enforced minimum disk size requirement for each Update Agent mode attemptpeering.
- **updateId** UniqueminFileSizePolicy The minimum content file size policy to allow the download using peering with delivery optimization. peerID The ID for eachthis delivery optimization client. scenarioID The ID of the scenario. sessionID The ID for the file download session. updateID The ID of the updatebeing downloaded.usedMemoryStream Indicates whether the download used memory streaming.

Microsoft.WindowsOSG.UpdateDU.DeviceUpdateAgentDeliveryOptClient.UpdateAgentInstallFailureCdnCommunication

This event collects informationrepresents a failure to download from a CDN with Delivery Optimization. It's used to understand and address problems regardingdownloads. The following fields are available: cdnHeaders The HTTP headers returned by the installphaseCDN. cdnIp The IP address of the newCDN. cdnUrl The URL of the CDN. clientTelId A random number used for device manifest UUP (Unified Update Platform) update scenariosampling. errorCode The error code that was returned. errorCount The total number of times this error code was seen since the last FailureCdnCommunication event was encountered. experimentId When running a test, whichthis is used to installcorrelate with other events that are part of the same test. fileID The ID of the file being downloaded. httpStatusCode The HTTP status code returned by the CDN. 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. sessionID The ID of the download session. Microsoft.OSG.DU.DeliveryOptClient.JobError This event represents a device manifest describing a setWindows Update job error. It allows for investigation of driver packagestop errors.

The following fields are available:

clientTelId A random number used for device sampling.

- **errorCode** The error code returned for. experimentId When running a test, this is used to correlate with other events that are part of the current install phase same test.
- **flightIdfileID** The unique identifier for each flightID of the file being downloaded.
- **objectIdjobID** The unique identifier for each diagnostics sessionWindows Update job ID.
- **relatedCV** Correlation vector value generated fromWindows Update events Microsoft.Windows.Update.DataMigrationFramework.DmfMigrationCompleted This event sends data collected at the latestUSO scan.
- **result** Outcomeend of the install phaseData Migration Framework (DMF) and parameters involved in its invocation, to help keep Windows up to date. The following fields are available: MigrationDurationInMilliseconds How long the DMF migration took (in milliseconds) MigrationEndTime A system timestamp ofwhen the updateDMF migration completed.
- **scenarioId** The unique identifierRevisionNumbers A collection of revision numbers for the update scenarioupdates associated with the DMF session.
- **sessionId** The unique identifierUpdateIds A collection of GUIDs for each updateupdates that are associated with the DMFsession.
- **updateIdWuClientId** The unique identifierGUID of the Windows Update client responsible for each update.triggering the DMF migration

Microsoft.Windows.Update.DeviceUpdateAgentDataMigrationFramework.UpdateAgentModeStartDmfMigrationStarted

This event sends datacollected 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: MigrationMicrosoftPhases Revision numbers for the startupdates that were installed. MigrationOEMPhases WU Update IDs for the updates that were installed. MigrationStartTime The timestamp representing the beginning of each mode during the processDMF migration RevisionNumbers A collection of updatingdevice manifest assets via the UUP (Unifiedrevision numbers associated with the UpdateIds. UpdateIds A collection of GUIDs identifying the upgrades that are running. WuClientId The GUID of the Windows Update Platform) update scenario, whichclient invoking DMF Microsoft.Windows.Update.DataMigrationFramework.MigratorResult This event sends DMF migrator data to help keep Windows up to date. The following fields are available: CurrentStep This is used to installthe last step the migrator reported before returning aresult. This tells us how far through the individual migrator the device manifest describing a setwas before failure. ErrorCode The result (as an HRESULT) of driver packagesthe migrator that just completed.MigratorId A GUID identifying the migrator that just completed. MigratorName The name of the migrator that just completed. RunDurationInSeconds The time it took for the migrator to complete. TotalSteps Migrators report progress in number of completed steps against the total steps. This is the total number of steps. Microsoft.Windows.Update.NotificationUx.DialogNotificationToBeDisplayed
This event indicates that a notification dialog box is about to be displayed to user.

The following fields are available:

- **flightIdAcceptAutoModeLimit** The unique identifiermaximum number of days for each flighta device to automatically enter Auto Reboot mode.
- **modeAutoToAutoFailedLimit** Themaximum number of days for Auto Reboot mode thatto fail before the RebootFailed dialog box is startingshown.
- **objectIdDeviceLocalTime** The unique valuelocal time on the device sending the event. EngagedModeLimit The number of days to switch between DTE dialog boxes. EnterAutoModeLimit The maximum number of days for each diagnostics sessiona device to enter Auto Reboot mode.
- **relatedCV** Correlation vectorETag OneSettings versioning value generated from the latest USO scan.
- **scenarioId** The scenario IDIsForcedEnabled Indicates whether Forced Reboot mode is enabled for this device. Example:MobileUpdateIsUltimateForcedEnabled Indicates whether Ultimate Forced Reboot mode is enabled for this device. NotificationUxState Indicates which dialog box is shown. NotificationUxStateString Indicates which dialog box is shown. RebootUxState Indicates the state of the restart (Engaged, DesktopLanguagePackAuto, DesktopFeatureOnDemandForced, or DesktopDriverUpdateUltimateForced). RebootUxStateString Indicates the state of the restart (Engaged, Auto, Forced, or UltimateForced). RebootVersion Version of DTE.

- **sessionId** Unique value for each Update AgentSkipToAutoModeLimit The minimum length of time to pass in restart pending before a device can be put into auto mode attempt.
- **updateId** Unique identifier for eachUpdateId The ID of the updatethat is pending restart to finish installation.UpdateRevision The revision of the update that is pending restart to finish installation.

Microsoft.Windows.Update.NotificationUx.EnhancedEngagedRebootAcceptAutoDialog This event indicates that the Enhanced Engaged restart "accept automatically" dialog box was displayed. The following fields are available: DeviceLocalTime The local time on the device sending the event. ETag OneSettings versioning value. ExitCode Indicates how users exited the dialog box. RebootVersion Version of DTE. UpdateId The ID of the update that is pending restart to finish installation. UpdateRevision The revision of the update that is pending restart to finish installation. UserResponseString The option that user chose on this dialog box. Microsoft.Windows.Update.NotificationUx.EnhancedEngagedRebootFirstReminderDialog

This event indicates that the Enhanced Engaged restart "first reminder" dialog box was displayed..

The following fields are available:

- **DeviceLocalTime** The local time on the device sending the event.
- **ETag** OneSettings versioning value.
- **ExitCode** Indicates how users exited the dialog box.
- **RebootVersion** Version of DTE.
- **UpdateId** The ID of the update that is pending restart to finish installation.
- **UpdateRevision** The revision of the update that is pending restart to finish installation.
- **UserResponseString** The option that user chose in this dialog box.
- **UtcTimeMicrosoft.Windows.Update.NotificationUx.EnhancedEngagedRebootForcedPrecursorDialog This event indicates that the Enhanced Engaged restart "forced precursor" dialog box was displayed.** Thefollowing fields are available: DeviceLocalTime The local timeon the device sending the event. ETag OneSettings versioning value. ExitCode Indicates how users exited the dialog box. RebootVersion Version of DTE. UpdateId The ID of the update thatis pending restart to finish installation. UpdateRevision The revision of theupdate that is pending restart to finish installation. UserResponseString The option that the user chose in this dialog box. Microsoft.Windows.Update.NotificationUx.EnhancedEngagedRebootForcedWarningDialog This event indicates that the Enhanced Engaged "forced warning" dialog box was displayed,. The following fields are available: DeviceLocalTime The local time on the device sending the event. ETag OneSettings versioning value. ExitCode Indicates how users exited the dialog box. RebootVersion Version of DTE. UpdateId The ID of the update that is pending restart to finish installation. UpdateRevision The revision of the update that is pending restart to finish installation. UserResponseString The option that the user chose in Coordinated Universal Timethis dialog box.

Microsoft.Windows.Update.OrchestratorNotificationUx.BlockedByBatteryLevelEnhancedEngagedRebootRebootFailedDialog

This event indicates that Windows Update activitythe Enhanced Engaged restart "restart failed" dialog box was blocked due to lowbattery leveldisplayed.

The following fields are available:

- **batteryLevelDeviceLocalTime** The current battery charge capacitylocal time of the device sending the event.
- **batteryLevelThresholdETag OneSettings versioning value. ExitCode Indicates how users exited the dialog box. RebootVersion Version of DTE. UpdateId** The battery capacity thresholdID of the update that is pending restart to stopupdate activityfinish installation.
- **updatePhaseUpdateRevision** The current staterevision of the update processthat is pending restart to finish installation.
- **wuDeviceid** DeviceUserResponseString The option that the user chose in this dialog box. Microsoft.Windows.Update.NotificationUx.EnhancedEngagedRebootRebootImminentDialog This event indicates that the Enhanced Engaged restart "restart imminent" dialog box was displayed. The following fields are available: DeviceLocalTime Time the dialog box was shown on the local device. ETag OneSettings versioning value. ExitCode Indicates how users exited the dialog box. RebootVersion Version of DTE. UpdateId The IDof the update that is pending restart to finish installation.UpdateRevision The revision of the update that is pending restart to finish installation. UserResponseString The option that user chose in this dialog box.

Microsoft.Windows.Update.OrchestratorNotificationUx.DTUCompletedWhenWuFlightPendingCommitEnhancedEngagedRebootSecondReminderDialog

This event indicates
that DTU completedthe second reminder dialog box was displayed for Enhanced Engaged restart. The following fields are available: DeviceLocalTime The time the dialog box was shown on the local device. ETag OneSettings versioning value. ExitCode Indicates how users exited the dialog box. RebootVersion Version of DTE. UpdateId The ID of the update that is pending restart to finish installation. UpdateRevision The revision of
the electronic software delivery (ESD), whenupdate that is pending restart to finish installation. UserResponseString The option that the user chose in this dialog box. Microsoft. Windows.Update.NotificationUx.EnhancedEngagedRebootThirdReminderDialog This event indicates that the third reminder dialog box for Enhanced Engaged restart was already in Pending Commit phasedisplayed. The following fields are available: DeviceLocalTime The time the dialog box was shown on the local device. ETag OneSettings versioning value. ExitCode Indicates how users exited the dialog box. RebootVersion Version ofDTE. UpdateId The ID of the feature updatethat is pending restart to finish installation.UpdateRevision The revision of the update that is pending restart to finish installation. UserResponseString The option that the user chose in this dialog box. Microsoft.Windows.Update.Orchestrator.CommitFailed
This event indicates that a device was unable to restart after an update.

The following fields are available:

errorCode The error code that was returned.

- **wuDeviceid** Device ID used byThe Windows Updatedevice GUID.

Microsoft.Windows.Update.Orchestrator.DTUEnabledDetection

This event indicates
that Inbox DTU functionalitya scan for a Windows Update occurred. The following fields are available: deferReason Reason why the device could not check for updates. detectionBlockreason Reason for detection not completing. detectionDeferreason A log of deferral reasons for every update state. errorCode The returned error code. 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 enabledcancelled, succeeded, or failed.flightID The specific ID of the Windows Insider build the device is getting. interactive Indicates whether the session was user initiated. revisionNumber Update revision number. updateId Update ID. updateScenarioType The update session type. wuDeviceid Unique device ID used by Windows Update. 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:

deferReason Reason for download not completing. detectionDeferreason Reason for download not completing errorCode An error code represented as a hexadecimal value. eventScenario End-to-
end update session ID. flightID The specific ID of the Windows Insider build the device is getting. interactive Indicates whether the session is user initiated. revisionNumber Update revision number. updateId Update ID. updateScenarioType The update session type.

- **wuDeviceid** DeviceUnique device ID used by Windows Update.

Microsoft.Windows.Update.Orchestrator.DTUInitiatedFlightInapplicable

This event indicates
that Inbox DTU functionalitythe update is no longer applicable to this device. The following fields are available: EventPublishedTime Time when this event was intiatedgenerated.flightID The specific ID of the Windows Insider build. revisionNumber Update revision number. updateId Unique Windows Update ID. updateScenarioType Update session type. UpdateStatus Last status of update. wuDeviceid Unique Device ID. 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:

- **dtuErrorCode** Return code from creating~~EventPublishedTime Time of~~ the DTU Com Server~~event~~.
- **isDtuApplicable** Determination~~flightID Unique update ID interactive Indicates whether the reboot initiation stage of the update process was entered as a result of user action. rebootOutsideOfActiveHours Indicates~~ whether DTU is applicable~~the reboot was to occur outside of active hours. revisionNumber Revision number of~~ the machine it is running on~~update.updateId Update ID. updateScenarioType The update session type. uxRebootstate Indicates the exact state of the user experience at the time the required reboot was initiated.~~
- **wuDeviceid** Device~~Unique device~~ ID used by Windows Update.

Microsoft.Windows.Update.Orchestrator.FailedToAddTimeTriggerToScanTask~~Install~~

This event indicated that USO failed to add~~sends launch data for~~ a trigger time~~Windows Update install~~ to a task~~help keep Windows up to date~~.

The following fields are available:

~~batteryLevel Current battery capacity in mWh or percentage left. deferReason Reason for install not completing.~~

- **errorCode** The Windows Update error code~~reppresented by a hexadecimal value~~.
- **wuDeviceid~~eventScenario End-to-end update session ID. flightID~~** The~~specific ID of the~~ Windows~~Insider build the device is getting. flightUpdate Indicates whether the update is a Windows Insider build. ForcedRebootReminderSet A boolean value that indicates if a forced reboot will happen for updates. installCommitfailedtime The time it took for a reboot to happen but the upgrade failed to progress. installRebootinitiatetime The time it took for a reboot to be attempted. interactive Identifies if session is user initiated. minutesToCommit The time it took to install updates. rebootOutsideOfActiveHours Indicates whether a reboot is scheduled outside of active hours. revisionNumber~~ Update~~revision number. updateId Update ID. updateScenarioType The update session type. 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. wuDeviceid Unique~~ device ID~~used by Windows Update~~.

Microsoft.Windows.Update.Orchestrator.StickUpdate~~PostInstall~~

This event is sent when the update service orchestrator (USO) indicates the update cannot be superseded by~~after~~ a newer~~Windows~~update~~install completes~~.

The following fields are available:

- **updateId~~batteryLevel Current battery capacity in mWh or percentage left. bundleId~~** Identifier associated with the specific piece~~content bundle. bundleRevisionnumber Identifies the revision number of~~ the content~~bundle.errorCode The error code returned for the current phase. eventScenario State of update action. flightID Unique update ID. sessionType The Windows Update session type (Interactive or Background).~~
- **wuDeviceid** Unique device ID controlled~~used~~ by the software distribution client~~Windows Update~~.

Microsoft.Windows.Update.Orchestrator.TerminatedByActiveHours~~RebootFailed~~

This event indicates that~~sends information about whether an~~ update activity was stopped due~~required a reboot and reasons for failure,~~ to active hours starting~~help keep Windows up to date~~.

The following fields are available:

- **activeHoursEnd~~batteryLevel Current battery capacity in mWh or percentage left. deferReason Reason for install not completing. EventPublishedTime~~** The end~~time that the reboot failure occurred. flightID Unique update ID. installRebootDeferreason Reason for reboot not occurring. rebootOutsideOfActiveHours Indicates whether a reboot was scheduled outside of~~ the active hours window.

- **activeHoursStartRebootResults Hex code indicating failure reason. Typically, we expect this to be a specific USO generated hex code. revisionNumber Update revision number. updateId Update ID. updateScenarioType** The startupdate session type. uxRebootstate Indicates the exact state of theuser 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. wuDeviceid Unique device ID used by Windows Update. 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. revisionNumber Update revision number. updateId Update ID. wuDeviceid Device ID for the device on which the reboot is restored. 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. rebootOutsideOfActiveHours Indicates whether a reboot is scheduled outside of active hours window.
- **updatePhaserevisionNumber Update revision number. systemNeededReason List of apps or tasks that are preventing the system from restarting. updateId Update ID. updateScenarioType** The currentupdate session type. uxRebootstate Indicates the exact state of theuser experience at the time the required reboot was initiated to ensure the correct update processand experience is provided to keep Windows up to date.
- **wuDeviceid** TheUnique device identifierID used by Windows Update.

Microsoft.Windows.Update.Orchestrator.TerminatedByBatteryLevelUpdatePolicyCacheRefresh

This
event is sentsends data on whether Update Management Policies were enabled on a device, to help keep Windows up to date. The following fields are available: configuredPoliciescount Number of policies on the device. policiesNamevaluesource Policy name and source of policy (group policy, MDM or flight). policyCacherefreshtime Time when update activitypolicy cache was stopped duetorefreshed. updateInstalluxsetting Indicates whether a low battery leveluser has set policies via a user experience option.wuDeviceid Unique device ID used by Windows Update. 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:

- **batteryLevelflightID** The current battery charge capacityspecific ID of the Windows Insider build the device is getting.
- **batteryLevelThresholdinteractive Indicates whether the reboot initiation stage of the update process was entered as a result of user action. revisionNumber Update revision number. updateId Update ID. updateScenarioType** The batterycapacity threshold to stop update activitysession type.
- **updatePhase** The currentuxRebootstate Indicates the exact state of theuser experience at the time the required reboot was initiated to ensure the correct update processand experience is provided to keep Windows up to date.
- **wuDeviceidUnique device ID used by Windows Update. Microsoft.Windows.Update.UpdateStackServicing.CheckForUpdates This event sends data about the UpdateStackServicing check for updates, to help keep Windows up to date.** Thefollowing fields are available: BspVersion The version of the BSP. CallerApplicationName The name of the USS scheduled task. Example: UssScheduled or UssBoot ClientVersion The version of the client. CommercializationOperator The name of the operator. DetectionVersion The string returned from the GetDetectionVersion export of the downloaded detection DLL. DeviceName The name of the device identifier.EventInstanceID The USS session ID. EventScenario The scenario of the event. Example: Started, Failed, or Succeeded OemName The name of the manufacturer. ServiceGuid The GUID of the service. StatusCode The HRESULT code of the operation. WUDeviceID The Windows Update device ID.

Microsoft.Windows.Update.OrchestratorUx.UnstickUpdateMusNotification.RebootNoLongerNeeded

This event is sent
when thea security update service orchestrator (USO) indicateshas successfully completed. The following fields are available: UtcTime The Coordinated Universal Time that
the update can be superseded byrestart was no longer needed. Microsoft.Windows.Update.Ux.MusNotification.RebootScheduled This event sends data about a newer updaterequired reboot that is scheduled with no user interaction, to help keep Windows up to date.

The following fields are available:

- **updateId** Identifier associated withactiveHoursApplicable Indicates whether Active Hours applies on this device. forcedReboot True, if a reboot is forced on thedevice. Otherwise, this is False rebootArgument Argument for the reboot task. It also represents sp

ecific piecereboot related action. rebootOutsideOfActiveHours True, if a reboot is scheduled outside of contentactive hours.False, otherwise. rebootScheduledByUser True, if a reboot is scheduled by user. False, if a reboot is scheduled automatically. rebootState Current state of the reboot. revisionNumber Revision number of the OS. scheduledRebootTime Time scheduled for the reboot. updateId Identifies which update is being scheduled.

- **wuDeviceid** Unique device ID controlledused by the software distribution clientWindows Update.

Microsoft.Windows.Update.Ux.MusNotification.UxBrokerScheduledTaskToastDisplayedToScheduleReboot

This event is sent when MUSE broker schedules a tasktoast notification is shown to the user about scheduling a device restart.

The following fields are available:

- **TaskArgumentUtcTime** TheCoordinated 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: activeHoursApplicable Is the restart respecting Active Hours? forcedReboot True, if a reboot is forced on the device. Otherwise, this is False rebootArgument The arguments with whichthat are passed to the taskOS for the restarted. rebootOutsideOfActiveHours Was the restart scheduled outside of Active Hours? rebootScheduledByUser Was the restart scheduled by the user? If the value isfalse, the restart was scheduledby the device.
- **TaskName** NamerebootState The state of the taskrestart.revisionNumber The revision number of the OS being updated. scheduledRebootTime Time of the scheduled reboot updateId The Windows Update device GUID. wuDeviceid The Windows Update device GUID.

Winlogon events Microsoft.Windows.Security.Winlogon.SetupCompleteLogon This event signals the completion of the setup process. It happens only once during the first logon. F