Contents

Overview of privacy controls for Microsoft 365 Apps for enterprise

8/25/2021 • 8 minutes to read • Edit Online

> **NOTE**
> For a list of Office products covered by this privacy information, see Privacy controls available for Office products.

Microsoft is committed to providing you with the information and controls you need to make choices about how your data is collected and used when you're using Microsoft 365 Apps for enterprise (previously named Office 365 ProPlus).

Starting with Version 1904 of Microsoft 365 Apps for enterprise, we are providing you with new, updated, and improved privacy controls for the following areas:

- ***Diagnostic data*** that is collected and sent to Microsoft about Office client software running on the user's device in your organization.
- ***Connected experiences*** that use cloud-based functionality to provide enhanced Office features to you and your users.

As part of these changes, there are new and updated user interface (UI) elements and policy settings.

Diagnostic data sent from Microsoft 365 Apps for enterprise to Microsoft

Diagnostic data is used to keep Office secure and up-to-date, detect, diagnose and remediate problems, and also make product improvements. This data does not include a user's name or email address, the content of the user's files, or information about apps unrelated to Office.

This diagnostic data is collected and sent to Microsoft about Office client software running on the user's device in your organization.

There are three levels of diagnostic data for Microsoft 365 Apps for enterprise client software that you can choose from:

- **Required** The minimum data necessary to help keep Office secure, up-to-date, and performing as expected on the device it's installed on.
- **Optional** Additional data that helps us make product improvements and provides enhanced information to help us detect, diagnose, and remediate issues.
- **Neither** No diagnostic data about Office client software running on the user's device is collected and sent to us. This option, however, significantly limits our ability to detect, diagnose, and remediate problems your users may encounter using Office.

Required diagnostic data could include, for example, information about the version of Office installed on the device or include information that indicates that Office applications are crashing when trying to open documents. Optional diagnostic data could include information about the time it takes to save a document, which could indicate an issue specific to saving to your device.

If you choose to send us optional diagnostic data, required diagnostic data is also included.

> **NOTE**
> Even if you choose Neither, required service data will be sent from the user's device to Microsoft. For more information, see Required service data for Office.

As an admin for your organization, you'll be able to use a policy setting to choose which level of diagnostic data is sent to us. Optional diagnostic data will be sent to Microsoft unless you change the setting. Providing optional diagnostic data better enables the Office engineering team at Microsoft to detect, diagnose, and mitigate issues to reduce impacts to your organization.

Your users won't be able to change the diagnostic data level for their devices if they are signed in to Office with their organizational credentials, which is sometimes referred to as a work or school account.

This diagnostic data doesn't include names of users, their email addresses, or the content of their Office files. Our system creates a unique ID that it associates with your user's diagnostic data. When we receive diagnostic data showing that one of our apps crashed 100 times, this unique ID lets us determine if it was a single user who crashed 100 times or if it was 100 different users who each crashed once. We don't use this unique ID to identify a specific user.

To see what diagnostic data is being sent to Microsoft, you can use the Diagnostic Data Viewer, which you can download and install for free from the Microsoft Store.

For more information, see the following articles:

- Required diagnostic data for Office
- Optional diagnostic data for Office
- Use policy settings to manage privacy controls for Microsoft 365 Apps for enterprise
- Use preferences to manage privacy controls for Office for Mac
- Use preferences to manage privacy controls for Office on iOS devices
- Use policy settings to manage privacy controls for Office on Android devices
- Using the Diagnostic Data Viewer with Office

Connected experiences for Microsoft 365 Apps for enterprise

Microsoft 365 Apps for enterprise consists of client software applications and connected experiences designed to enable you to create, communicate, and collaborate more effectively. Working with others on a document stored on OneDrive for Business or translating the contents of a Word document into a different language are examples of connected experiences.

We understand that you might want to choose which types of connected experiences are available to your users when working in Office applications. As an admin for your organization, you'll have policy settings that allow you to choose whether to provide the following types of connected experiences to your users:

- **Experiences that analyze your content** Experiences that use your Office content to provide you with design recommendations, editing suggestions, data insights, and similar features. For example, PowerPoint Designer or Translator.
- **Experiences that download online content** Experiences that allow you to search and download online content including templates, images, 3D models, videos, and reference materials to enhance your documents. For example, Office templates or PowerPoint QuickStarter.

For example, you might choose to provide your users with connected experiences that download online content, but not connected experiences that analyze content. If you don't configure these policy settings, all these connected experiences will be available to your users.

In addition, there is a policy setting that allows you turn off all these connected experiences, and which will also turn off other connected experiences, such as document co-authoring and online file storage. But even if you use this policy setting to turn off all these connected experiences, certain Office functionality will remain available, such as synching your mailbox in Outlook, using Teams or Skype for Business, as well as the essential services described below.

If you choose not to provide your users with certain types of connected experiences, either the ribbon or menu command for those connected experiences will be grayed out or the users will get an error message when they try to use those connected experiences.

Your users won't be able to choose whether to turn these connected experiences on or off if they are signed in to Office with their organizational credentials, which is sometimes referred to as a work or school account.

For more information, see the following articles:

- Connected experiences in Office
- Use policy settings to manage privacy controls for Microsoft 365 Apps for enterprise
- Use preferences to manage privacy controls for Office for Mac
- Use preferences to manage privacy controls for Office on iOS devices
- Use policy settings to manage privacy controls for Office on Android devices

Optional connected experiences for Microsoft 365 Apps for enterprise

In addition to the connected experiences mentioned above that are included with Microsoft 365 Apps for enterprise, there are optional connected experiences that you may choose to allow your users to access with their organization account. For example, the LinkedIn features of the Resume Assistant in Word or the 3D Maps feature in Excel, which uses Bing.

These are optional connected experiences that are not covered by your organization's commercial agreement with Microsoft but are governed by separate terms and conditions. Optional connected experiences offered by Microsoft directly to your users are governed by the Microsoft Services Agreement instead of the Online Services Terms.

Because these optional connected experiences are governed by separate terms and conditions, you manage them separately from the connected experiences mentioned above. As an admin for your organization, you'll be able to use a policy setting to choose whether to make these optional connected experiences available, as a group, to your users. If you don't configure this policy setting, these optional connected experiences are available to your users.

Even if you choose to make these optional connected experiences available to your users, your users will have the option to turn these optional connected experiences off as a group by going to the privacy settings dialog box. Your users will only have this choice if they are signed in to Office with their organizational credentials (sometimes referred to as a work or school account), not if they are signed in with a personal email address.

For more information, see the following articles:

- Overview of optional connected experiences in Office
- Use policy settings to manage privacy controls for Microsoft 365 Apps for enterprise
- Use preferences to manage privacy controls for Office for Mac
- Use preferences to manage privacy controls for Office on iOS devices
- Use policy settings to manage privacy controls for Office on Android devices
- Use policy settings to manage privacy controls for Office for the web applications

Required service data for connected experiences

As you use a connected experience, data is sent to and processed by Microsoft to provide you that connected experience. This data is crucial because this information enables us to deliver these cloud-based connected experiences. We refer to this data as required service data.

Required service data can include information related to the operation of the connected experience that is needed to keep the underlying service secure, up to date, and performing as expected. If you choose to use a connected experience that analyzes your content, for example Translate in Word, the text you typed and selected to translate in the document is also sent and processed to provide you the connected experience. Required service data can also include information needed by a connected experience to perform its task, such as configuration information about the Office app.

For more information, see Required service data for Office.

Essential services for Microsoft 365 Apps for enterprise

There is also a set of services that are essential to how Microsoft 365 Apps for enterprise functions and cannot be disabled. For example, the licensing service that confirms that you are properly licensed to use Microsoft 365 Apps for enterprise. Required service data about these services is collected and sent to Microsoft, regardless of any other policy settings that you have configured.

For more information, see Essential services for Office.

Related articles

- Privacy at Microsoft
- Windows privacy

Privacy controls available for Office products

8/25/2021 • 6 minutes to read • Edit Online

Microsoft is committed to providing you with the information and controls you need to make informed choices about how your data is collected and used when you're using Office products. This includes when you use the Office products on Windows, Mac, iOS, and Android devices as well as when you use the Office apps on the web.

Starting in April 2019, we began providing new, updated, and improved privacy controls for diagnostic data and connected experiences to our most current Office products. For more information, see Overview of privacy controls for Microsoft 365 Apps for enterprise.

The sections in this article contain the following information:

- Which Office products and versions provide our most recent privacy controls
- Which of those privacy controls are available in those Office products and versions
- Links to articles with more information about managing these privacy controls

The sections are organized by where Office is being run, such as on a Windows or Mac device or from a web browser.

Office on Windows devices

Our most recent privacy controls are available for the following Office products when using Office on Windows devices:

- Desktop versions of the Office applications that come with Microsoft 365 or Office 365 subscription plans. For example, the Office apps that come with the Microsoft 365 Personal plan (for home), the Microsoft 365 Business Standard plan (for businesses), or the Microsoft 365 E5 plan (for enterprises).
- Retail versions of Office 2019 and Office 2016, which are available as a one-time purchase. For example, Office Professional 2019 or Office Home & Student 2016.
- Desktop versions of Project and Visio that come with some subscription plans, such as Project Plan 5 or Visio Plan 2.
- Retail versions of Project 2019, Project 2016, Visio 2019, and Visio 2016, which are available as a one-time purchase.

For Windows devices, the following table lists the apps and the minimum version of those apps which have our most recent privacy controls.

APP	MINIMUM VERSION
Access	1904
Excel	1904
OneDrive	20.084.0426.0007
OneNote	1904
Outlook	1904

APP	MINIMUM VERSION
PowerPoint	1904
Project	1904
Publisher	1904
Skype for Business	16.0.11629
Teams	1.3.00.13565
Visio	1904
Word	1904

For those versions of Office apps on Windows devices, the following privacy controls are available:

- Configure the level of client software diagnostic data sent by Office to Microsoft
- Allow the use of connected experiences in Office that analyze content
- Allow the use of connected experiences in Office that download online content
- Allow the use of additional optional connected experiences in Office
- Allow the use of connected experiences in Office

Some Office products might not have certain types of connected experiences, so some privacy controls won't be relevant.

To configure these privacy controls for your users that are using Office on Windows devices in your organization, you can use Group Policy or the Office cloud policy service. For more information, see Use policy settings to manage privacy controls for Microsoft 365 Apps for enterprise.

> **NOTE**
>
> The "Allow the use of additional optional connected experiences in Office" privacy control is supported on the following *volume licensed* versions of Office, Project, and Visio.
>
> - Office 2019, Project 2019, and Visio 2019, when they're configured to use the PerpetualVL2019 update channel. In that case, you must be using at least Version 1808 (Build 10367.20048), which was released on October 13, 2020.
> - Project 2019 and Visio 2019, when they're configured to use an update channel other than PerpetualVL2019. For example, if they're configured to use Current Channel, Monthly Enterprise Channel, or Semi-Annual Enterprise Channel. In that case, you must be using at least Version 1904.
> - Project 2016 and Visio 2016 that are installed by using Click-to-Run instead of being installed by using Windows Installer (MSI). In that case, you must be using at least Version 1904.

Office on Mac devices

Our most recent privacy controls are available for the following Office products when using Office on Mac devices:

- Office for Mac, which is available with any Office 365 or Microsoft 365 subscription plan that includes the desktop versions of the Office apps. For example, the Office apps that come with the Microsoft 365 Family plan (for home), the Microsoft 365 Business Premium plan (for businesses), or the Microsoft 365 E3 plan (for enterprises).
- Office 2019 for Mac, which is available as a one-time purchase from a retail store or through a volume

licensing agreement.

For Mac devices, the following table lists the apps and the minimum version of those apps which have our most recent privacy controls.

APP	MINIMUM VERSION
Excel	16.28
OneDrive	20.169.0823.0003
OneNote	16.28
Outlook	16.28
PowerPoint	16.28
Skype for Business	16.28.0.192
Teams	1.3.00.9221
Word	16.28

For those versions of Office apps on Mac devices, the following privacy controls are available:

- Configure the level of client software diagnostic data sent by Office to Microsoft
- Allow the use of connected experiences in Office that analyze content
- Allow the use of connected experiences in Office that download online content
- Allow the use of additional optional connected experiences in Office
- Allow the use of connected experiences in Office

Some Office products might not have certain types of connected experiences, so some privacy controls won't be relevant.

To configure these privacy controls for your users that are using Office on Mac devices in your organization, you can use preferences or the Office cloud policy service. For more information, see Use preferences to manage privacy controls for Office for Mac.

Office on iOS devices

Our most recent privacy controls are available for the following Office products when using Office on iOS devices:

APP	MINIMUM VERSION
Excel	2.30
Lens	2.38
Office app	2.34
OneDrive	11.19.11
OneNote	16.30

APP	MINIMUM VERSION
Outlook	4.30.0
Planner	1.9.4
PowerPoint	2.30
Skype for Business	6.26.2
Teams	1417/2.0.18(2020072902)
Visio Viewer	1.17
Word	2.30

For those versions of Office apps on iOS devices, the following privacy controls are available:

- Configure the level of client software diagnostic data sent by Office to Microsoft
- Allow the use of connected experiences in Office that analyze content
- Allow the use of connected experiences in Office that download online content
- Allow the use of additional optional connected experiences in Office

Some Office products might not have certain types of connected experiences, so some privacy controls won't be relevant.

To configure these privacy controls for your users that are using Office on iOS devices in your organization, you can use preferences or the Office cloud policy service. For more information, see Use preferences to manage privacy controls for Office on iOS devices.

Office on Android devices

Our most recent privacy controls are available for the following Office products when using Office on Android devices:

APP	MINIMUM VERSION
Excel	16.0.12228.20260
Lens	16.0.12730.20080
Office app	16.0.12430.20254
OneDrive	5.47
OneNote	16.0.12228.20004
Outlook	4.1.71
Planner	1.12.0
PowerPoint	16.0.12228.20260

APP	MINIMUM VERSION
Skype for Business	6.27.0.12
Teams	1416/1.0.0.2020080601
Word	16.0.12228.20260

For those versions of Office apps on Android devices, the following privacy controls are available:

- Configure the level of client software diagnostic data sent by Office to Microsoft
- Allow the use of connected experiences in Office that analyze content
- Allow the use of connected experiences in Office that download online content
- Allow the use of additional optional connected experiences in Office

Some Office products might not have certain types of connected experiences, so some privacy controls won't be relevant.

To configure these privacy controls for your users that are using Office on Android devices in your organization, you need to use the Office cloud policy service. For more information, see Use policy settings to manage privacy controls for Office on Android devices.

Office for the web

Our most recent privacy controls are available for the following Office products when using Office from a web browser:

- Excel
- OneNote
- PowerPoint
- Visio
- Word

There is no minimum version listed for these Office apps because Microsoft manages which version is available.

For Office on the web, the following privacy controls are available:

- Allow the use of additional optional connected experiences in Office

To configure this privacy control for your users that are using Office on the web, you need to use the Office cloud policy service. For more information, see Use policy settings to manage privacy controls for Office for the web applications.

Related articles

- Privacy and Microsoft Teams
- Privacy settings in Microsoft Whiteboard
- What version of Office am I using?
- What version of Outlook do I have?

Use policy settings to manage privacy controls for Microsoft 365 Apps for enterprise

8/25/2021 • 14 minutes to read • Edit Online

NOTE

For a list of Office products covered by this privacy information, see Privacy controls available for Office products.

Microsoft is committed to providing you with the information and controls you need to make choices about how your data is collected and used when you're using Microsoft 365 Apps for enterprise (previously named Office 365 ProPlus).

Starting with Version 1904 of Microsoft 365 Apps for enterprise, there are new policy settings that will allow you to control settings related to the following:

- ***Diagnostic data*** that is collected and sent to Microsoft about Office client software being used
- ***Connected experiences*** that use cloud-based functionality to provide enhanced Office features to you and your users.

The following are the five new policy settings:

- Configure the level of client software diagnostic data sent by Office to Microsoft
- Allow the use of connected experiences in Office that analyze content
- Allow the use of connected experiences in Office that download online content
- Allow the use of additional optional connected experiences in Office
- Allow the use of connected experiences in Office

These policy settings can be implemented by using either Group Policy or the Office cloud policy service. If you're using Group Policy, you need to download the most current version of the Administrative Template files (ADMX/ADML) from the Microsoft Download Center.

NOTE

- For information on how to manage privacy controls for Office for Mac, see Use preferences to manage privacy controls for Office for Mac.
- For information about similar settings for Office on iOS devices, see Use preferences to manage privacy controls for Office on iOS devices.
- For information about similar settings for Office on Android devices, see Use policy settings to manage privacy controls for Office on Android devices.
- For information about privacy controls for Office for the web applications, see Use policy settings to manage privacy controls for Office for the web applications.

If you're using the Group Policy Management tool, all these policy settings are located under User Configuration\Policies\Administrative Templates\Microsoft Office 2016\Privacy\Trust Center.

These new policy settings also apply to the desktop versions of Project and Visio that come with some subscription plans, such as Project Plan 5 or Visio Plan 2. They also apply to Microsoft 365 Apps for business (previously named Office 365 Business).

There are also some existing policy settings that will no longer apply to Microsoft 365 Apps for enterprise, and there are some user interface (UI) changes for privacy settings that you should be aware of because your users might notice those changes and ask about them.

As with any new policy settings, you should carefully test them out in a limited, controlled environment to ensure the settings you configure have the desired effect before you implement the policy settings more widely in your organization.

Policy setting for diagnostic data

Diagnostic data is used to keep Office secure and up-to-date, detect, diagnose and remediate problems, and also make product improvements.

You can use the *Configure the level of client software diagnostic data sent by Office to Microsoft* policy setting to choose what level of diagnostic data is sent to Microsoft.

If you enable this policy setting, you must choose which level of diagnostic data is sent to Microsoft. Your choices are Required, Optional, or Neither.

- If you choose ***Required***, the minimum data necessary to help keep Office secure, up-to-date, and performing as expected on the device it's installed on is sent to Microsoft.
- If you choose ***Optional***, additional data that helps make product improvements and provides enhanced information to help detect, diagnose, and remediate issues is sent to Microsoft. If you choose to send optional diagnostic data, required diagnostic data is also included.
- If you choose ***Neither***, no diagnostic data about Office client software running on the user's device is sent to Microsoft. This option, however, significantly limits Microsoft's ability to detect, diagnose, and remediate problems that your users may encounter when using Office.

If you disable or don't configure this policy setting, both optional and required diagnostic data are sent to Microsoft.

> **NOTE**
> Even if you choose ***Neither***, required service data will be sent from the user's device to Microsoft. For more information, see Required service data for Office.

For more information about diagnostic data, see the following:

- Overview of privacy controls for Microsoft 365 Apps for enterprise
- Required diagnostic data for Office
- Optional diagnostic data for Office
- Using the Diagnostic Data Viewer with Office

Policy settings for connected experiences

Microsoft 365 Apps for enterprise consists of client software applications and connected experiences designed to enable you to create, communicate, and collaborate more effectively. Working with others on a document stored on OneDrive for Business or translating the contents of a Word document into a different language are examples of connected experiences.

We understand that you might want to choose which types of connected experiences are available to your users when they're working in Office applications. So we have provided four new policy settings for you:

- Allow the use of connected experiences in Office that analyze content

- Allow the use of connected experiences in Office that download online content
- Allow the use of additional optional connected experiences in Office
- Allow the use of connected experiences in Office

If you don't configure these policy settings, all connected experiences are available. This gives your users all the features and functionality accessible through Microsoft 365 Apps for enterprise. But we understand that you might need to turn off some or all of these connected experiences to meet certain requirements of your organization.

If you choose not to provide your users with certain types of connected experiences, either the ribbon or menu command for those connected experiences will be grayed out or users will get an error message when they try to use those connected experiences. In that case, no required service data for those connected experiences will be sent to Microsoft.

Your users won't be able to choose whether to turn these connected experiences included with Microsoft 365 Apps for enterprise on or off if they are signed into Office with their organizational credentials, which is sometimes referred to as a work or school account.

Policy setting for connected experiences that analyze your content

These are experiences that use your Office content to provide you with design recommendations, editing suggestions, data insights, and similar features. For example, PowerPoint Designer or Translator. For a list of these connected experiences, see Connected experiences in Office.

You can use the *Allow the use of connected experiences in Office that analyze content* policy setting to control whether these types of connected experiences are available to your users. If you don't configure this policy setting, these connected experiences will be available to your users.

Note that if you disable the *Allow the use of connected experiences in Office* policy setting, connected experiences that analyze content won't be available to your users.

Policy setting for connected experiences that download online content

These are experiences that allow you to search and download online content including templates, images, 3D models, videos, and reference materials to enhance your documents. For example, Office templates or PowerPoint QuickStarter. For a list of these connected experiences, see Connected experiences in Office.

You can use the *Allow the use of connected experiences in Office that download online content* policy setting to control whether these types of connected experiences are available to your users. If you don't configure this policy setting, these connected experiences will be available to your users.

Note that if you disable the *Allow the use of connected experiences in Office* policy setting, connected experiences that download online content won't be available to your users.

Policy setting for optional connected experiences

In addition to the connected experiences mentioned above that are included with Microsoft 365 Apps for enterprise, there are some optional connected experiences that you may choose to allow your users to access with their organization account. For example, the LinkedIn features of the Resume Assistant in Word or the 3D Maps feature in Excel, which uses Bing. For more examples, see Overview of optional connected experiences in Office.

These connected experiences are different because they are not covered by your organization's commercial agreement with Microsoft. Optional connected experiences are offered by Microsoft directly to your users and are governed by the Microsoft Services Agreement instead of the Online Services Terms. In some cases, third-party content or functionality are provided through these optional connected experiences and other terms may also apply. For more information, see Overview of optional connected experiences in Office.

You can use the *Allow the use of additional optional connected experiences in Office* policy setting to control

whether these types of connected experiences are available to your users. If you don't configure this policy setting, these optional connected experiences will be available to your users.

> **NOTE**
> To apply the *Allow the use of additional optional connected experiences in Office* policy setting to volume licensed versions of Office 2019, Project 2019, or Visio 2019, you must use Group Policy. You can't use the Office cloud policy service. This applies to when Office 2019, Project 2019, or Visio 2019 is configured to use the PerpetualVL2019 update channel.

Even if you choose to make these optional connected experiences available to your users, your users will have the option to turn them off as a group by going to the privacy settings dialog box. Your users will only have this choice if they are signed into Office with their organizational credentials (sometimes referred to as a work or school account), not if they are signed in with a personal email address.

Also, some of these optional connected experiences are also considered to be connected experiences that analyze content or that download online content. For example, Insert Online Pictures is an optional connected experience, powered by Microsoft Bing, but it's also considered to be a connected experience that downloads online content. Therefore, if you disable the *Allow the use of connected experiences in Office that download online content* policy setting, Insert Online Pictures won't be available to your users. It won't be available even if you have enabled the *Allow the use of additional optional connected experiences in Office* policy setting. For more information about which connected experiences analyze content or download online content, see Connected experiences in Office.

There is one exception to take note of. The *Allow the use of additional optional connected experiences in Office* policy setting does not control experiences that require you to connect your LinkedIn account to your Microsoft work or school account. To control these types of experiences, such as the LinkedIn information on a profile card in Outlook, see LinkedIn in Microsoft apps and services and Integrate LinkedIn account connections in Azure Active Directory.

Policy setting for most connected experiences

You can use the *Allow the use of connected experiences in Office* policy setting to control whether most connected experiences accessible through Microsoft 365 Apps for enterprise are available to your users. If you disable the policy setting, the following types of connected experiences won't be available to your users:

- Experiences that analyze your content
- Experiences that download online content
- Optional connected experiences

In addition, if you disable this policy setting, most other connected experiences are also turned off, such as co-authoring and online file storage. For a list of these other connected experiences, see Connected experiences in Office.

But even if you disable this policy setting, limited Office functionality will remain available, such as synching a mailbox in Outlook, and Teams and Skype for Business will continue to work. Essential services, such as the licensing service that confirms that you're properly licensed to use Office, will also remain available.

Existing policy settings that are replaced by new policy settings

There are two existing policy settings that are no longer applicable to Microsoft 365 Apps for enterprise, starting with Version 1904. Those policy settings are the following:

- **Send personal information**, which can be found under User Configuration\Policies\Administrative Templates\Microsoft Office 2016\Privacy\Trust Center.

- **Online Content Options**, which can be found under User Configuration\Policies\Administrative Templates\Microsoft Office 2016\Tools | Options | General | Service Options...\Online Content.

Starting with Version 1904, configuring these two existing policy settings will have no effect on Microsoft 365 Apps for enterprise. They are no longer applicable because their functionality is replaced by these new policy settings:

- Allow the use of connected experiences in Office that analyze content
- Allow the use of connected experiences in Office that download online content
- Allow the use of additional optional connected experiences in Office
- Allow the use of connected experiences in Office

These new policy settings can give you a finer level control than the two existing policy settings. For example, previously if you used the *Send personal information* policy setting, both PowerPoint QuickStarter and Smart Lookup would be turned off. But now, with the new policy settings, if you use the *Allow the use of connected experiences in Office that analyze content* policy setting to turn off that type of connected experiences, only Smart Lookup is turned off. PowerPoint QuickStarter will still be available to your users.

The policy settings still appear in the Group Policy Management tool because they are still applicable to volume licensed versions of Office 2016 and Office 2019, such as Office Professional Plus 2019.

What about existing policy settings that control connected experiences?

As you probably already know, there are some existing policy settings that allow you to control connected experiences. Here are a few examples of existing policy settings:

- *PowerPoint Designer Options*, under User Configuration\Policies\Administrative Templates\Microsoft Office 2016\Tools | Options | General | Service Options...\PowerPoint Designer
- *Turn off QuickStarter*, under User Configuration\Policies\Administrative Templates\Microsoft PowerPoint 2016\PowerPoint Options\General
- *Allow LinkedIn Resume Assistant feature*, under User Configuration\Policies\Administrative Templates\Microsoft Word 2016\Word Options\General

You can still use these existing policy settings if you want to turn off individual connected experiences. But keep in mind that if you use one of the new policy settings, that new policy setting might turn off a connected experience that you turned on by using a different policy setting. For example, if you enable the *Allow LinkedIn Resume Assistant feature* policy setting, but disable the *Allow the use of connected experiences in Office* policy setting, the LinkedIn Resume Assistant won't be available to your users.

In general, if one policy setting is configured to turn on a specific connected experience while at the same time another policy setting is configured to turn off that type of connected experience, then that specific connected experience is turned off for your users.

Privacy-related changes to the Office UI

There are some changes to the user interface (UI) of Microsoft 365 Apps for enterprise related to privacy that your users might notice and ask about. These changes are a direct result of the new privacy controls and policy settings available starting in Version 1904.

Dialog about optional connected experiences

If you have chosen to provide your users with optional connected experiences, the first time your users open an Office app after they've been updated to Version 1904 or later, an informational dialog box will appear. This dialog box informs your users that you have given them the choice to use these optional connected experiences

and lets them know they can go to **File** > **Account** > **Account Privacy** to change this setting.

Privacy settings removed from the Office UI

The following settings are removed from **File** > **Options** > **Trust Center** > **Trust Center Settings...** > **Privacy Options**:

- Get designs, information, recommendations, and services by allowing Office to access and make product improvements based on Office content on my device.
- Let Office connect to online services from Microsoft to provide functionality that's relevant to your usage and preference.

Also, under **File** > **Options** > **General**, the choice to enable Office intelligent services is removed.

As the admin for your organization, you now control the equivalent settings to these through the new policy settings described earlier.

Privacy settings added to the Office UI

The following are new elements added to the Office UI:

- Under **File** > **Account**, users will see a new choice for **Account Privacy** > **Manage Settings**. It's under **Manage Settings** where users can turn off optional connected experiences, if you have given them that option.
- Under **File** > **Options** > **Trust Center** > **Trust Center Settings...** > **Privacy Options,** there is an option to enable the use of the Diagnostic Data Viewer on the device.

Control privacy settings by editing the registry

Some admins prefer to change settings directly in the registry, for example by using a script, instead of by using Group Policy or the Office cloud policy service. You can use the following information to configure privacy settings directly in the registry.

POLICY SETTING	REGISTRY SETTING	VALUES
Configure the level of client software diagnostic data sent by Office to Microsoft	SendTelemetry	1=Required 2=Optional 3=Neither
Allow the use of connected experiences in Office that analyze content	UserContentDisabled	1=Enabled 2=Disabled
Allow the use of connected experiences in Office that download online content	DownloadContentDisabled	1=Enabled 2=Disabled
Allow the use of additional optional connected experiences in Office	ControllerConnectedServicesEnabled	1=Enabled 2=Disabled
Allow the use of connected experiences in Office	DisconnectedState	1=Enabled 2=Disabled

To create a .reg file for the privacy settings, open Notepad and copy in the following lines. Adjust the values to suit your needs, and then save the file. Be sure the file name has an extension of .reg

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Policies\Microsoft\office\16.0\common\privacy]
"disconnectedstate"=dword:00000001
"usercontentdisabled"=dword:00000001
"downloadcontentdisabled"=dword:00000001
"controllerconnectedservicesenabled"=dword:00000001

[HKEY_CURRENT_USER\Software\Policies\Microsoft\office\common\clienttelemetry]
"sendtelemetry"=dword:00000002
```

For example, you can use this .reg file with the regedit.exe command in a script to configure privacy settings for the user.

Use preferences to manage privacy controls for Office for Mac

8/25/2021 • 7 minutes to read • Edit Online

> **NOTE**
> For a list of Office products covered by this privacy information, see Privacy controls available for Office products.

Starting with Version 16.28 of Office for Mac, there are new preference settings that allow you to control settings related to the following:

- ***Diagnostic data*** that is collected and sent to Microsoft about Office client software being used.
- ***Connected experiences*** that use cloud-based functionality to provide enhanced Office features to you and your users.

In addition, there is a new preference setting related to a **Required Data Notice** dialog for Microsoft AutoUpdate (MAU).

For more information about diagnostic data and connected experiences, see Overview of privacy controls.

> **NOTE**
> - For information about similar settings for Office on computers running Windows, see Use policy settings to manage privacy controls for Microsoft 365 Apps for enterprise.
> - For information about similar settings for Office on iOS devices, see Use preferences to manage privacy controls for Office on iOS devices.

Setting preferences

These new preference settings are CFPreferences API compatible and can be set using the `defaults` command in Terminal, or enforced through a Configuration Profile or Mobile Device Management (MDM) server. When the preferences are enforced, the user cannot change the values, and any in-app controls will appear disabled.

> **NOTE**
> You can also use the Office cloud policy service and these 5 policy settings:
>
> - Configure the level of client software diagnostic data sent by Office to Microsoft
> - Allow the use of connected experiences in Office that analyze content
> - Allow the use of connected experiences in Office that download online content
> - Allow the use of additional optional connected experiences in Office
> - Allow the use of connected experiences in Office
>
> For more information on using the Office cloud policy service, see Overview of the Office cloud policy service.

Preference setting for diagnostic data

Diagnostic data is used to keep Office secure and up-to-date, detect, diagnose and remediate problems, and also

make product improvements. For more information, see Diagnostic data sent from Microsoft 365 Apps for enterprise to Microsoft.

CATEGORY	DETAILS
Preference Domain	`com.microsoft.office`
Key	`DiagnosticDataTypePreference`
Data Type	String
Possible values	`BasicDiagnosticData` *(this value sets the level to Required)* `FullDiagnosticData` *(this value sets the level to Optional)* `ZeroDiagnosticData` *(this value sets the level to Neither)*
Availability	16.28 and later

If you don't set this preference, both required and optional diagnostic data are sent to Microsoft if users with an Office 365 (or Microsoft 365) subscription are signed in with a work or school account or if users have a volume licensed version of Office 2019 for Mac. Also, these users can't change the level of diagnostic data regardless of how you set this preference.

For other users, such as home users with an Office 365 (or Microsoft 365) subscription, only required diagnostic data is sent, unless the user chooses to also send optional diagnostic data by going to **Preferences** > **Privacy**.

Preference setting for connected experiences that analyze your content

Connected experiences that analyze your content are experiences that use your Office content to provide you with design recommendations, editing suggestions, data insights, and similar features. For example, PowerPoint Designer or Researcher in Word. For a list of these connected experiences, see Connected experiences in Office.

CATEGORY	DETAILS
Preference Domain	`com.microsoft.office`
Key	`OfficeExperiencesAnalyzingContentPreference`
Data Type	Boolean
Possible values	`TRUE` *(enabled)* `FALSE` *(disabled)*
Availability	16.28 and later

If you don't set this preference, connected experiences that analyze content are available to users.

If the user has an Office 365 (or Microsoft 365) subscription and is signed in with a work or school account or if the user has a volume licensed version of Office 2019 for Mac, then the user can't turn off connected experiences that analyze content.

For other users, such as home users with an Office 365 (or Microsoft 365) subscription, the user can choose to turn off connected experiences that analyze content by going to **Preferences** > **Privacy**.

Preference setting for connected experiences that download online content

Connected experiences that download online content are experiences that allow you to search and download online content including templates, images, 3D models, videos, and reference materials to enhance your documents. For example, Office templates or PowerPoint QuickStarter. For a list of these connected experiences, see Connected experiences in Office.

CATEGORY	DETAILS
Preference Domain	`com.microsoft.office`
Key	`OfficeExperiencesDownloadingContentPreference`
Data Type	Boolean
Possible values	`TRUE` *(enabled)* `FALSE` *(disabled)*
Availability	16.28 and later

If you don't set this preference, connected experiences that download online content are available to users.

If the user has an Office 365 (or Microsoft 365) subscription and is signed in with a work or school account or if the user has a volume licensed version of Office 2019 for Mac, then the user can't turn off connected experiences that download online content.

For other users, such as home users with an Office 365 (or Microsoft 365) subscription, a user can choose to turn off connected experiences that download online content by going to **Preferences** > **Privacy**.

Preference setting for optional connected experiences

In addition to the connected experiences mentioned above, there are some optional connected experiences that you may choose to allow your users to access with their organization account, which is sometimes referred to as a work or school account. For example, the LinkedIn features of the Resume Assistant in Word or the Weather Bar in Outlook, which uses MSN Weather. For more examples, see Overview of optional connected experiences in Office.

CATEGORY	DETAILS
Preference Domain	`com.microsoft.office`
Key	`OptionalConnectedExperiencesPreference`
Data Type	Boolean
Possible values	`TRUE` *(enabled)* `FALSE` *(disabled)*
Availability	16.28 and later

If you don't set this preference, optional connected experiences are available to users with an Office 365 (or Microsoft 365) subscription that are signed in with a work or school account or users who have a volume

licensed version of Office 2019 for Mac. Unless you have set this preference to `FALSE`, these users can choose to turn off optional connected experiences by going to **Preferences** > **Privacy**.

For other users, such as home users with an Office 365 (or Microsoft 365) subscription, there isn't an option to turn off optional connected experiences.

Preference setting for most connected experiences

You can use this preference to control whether most connected experiences are available to your users.

CATEGORY	DETAILS
Preference Domain	`com.microsoft.office`
Key	`ConnectedOfficeExperiencesPreference`
Data Type	Boolean
Possible values	`TRUE` *(enabled)* `FALSE` *(disabled)*
Availability	16.28 and later

If you don't set this preference, all connected experiences are available to your users, unless you have set one of the other preferences for connected experiences previously mentioned, such as `OfficeExperiencesAnalyzingContentPreference`.

For example, if you set this preference to `FALSE`, the following types of connected experiences won't be available to your users:

- Experiences that analyze your content
- Experiences that download online content
- Optional connected experiences

In addition, if you set this preference to `FALSE`, most other connected experiences are also turned off, such as coauthoring and online file storage. For a list of these other connected experiences, see Connected experiences in Office.

But even if you set this preference to `FALSE`, limited Office functionality will remain available, such as synching a mailbox in Outlook, and Teams and Skype for Business will continue to work. Essential services, such as the licensing service that confirms that you're properly licensed to use Office, will also remain available.

If the user has an Office 365 (or Microsoft 365) subscription and is signed in with a work or school account or if the user has a volume licensed version of Office 2019 for Mac, then the user can't turn off most connected experiences.

For other users, such as home users with an Office 365 (or Microsoft 365) subscription, a user can choose to turn off most connected experiences by going to **Preferences** > **Privacy**.

Preference setting for the Required Data Notice dialog for Microsoft AutoUpdate

The first time Version 4.12 or later of Microsoft AutoUpdate (MAU) is launched, users will see a **Required Data Notice** dialog, which provides them with information about what data from MAU is sent to Microsoft.

If you don't want your users to see this **Required Data Notice** dialog for Microsoft AutoUpdate, you can set the following preference. Regardless of which value you set, the dialog won't be shown to your users.

CATEGORY	DETAILS
Preference Domain	`com.microsoft.autoupdate2`
Key	`AcknowledgedDataCollectionPolicy`
Data Type	String
Possible values	`RequiredDataOnly` `RequiredAndOptionalData`
Availability	4.12 and later

If you let your users see this dialog, then when the user chooses **OK**, the value `RequiredDataOnly` is written to `AcknowledgedDataCollectionPolicy` and the dialog is not shown to the user again.

Related articles

- Configuration Profile Reference (Apple developer documentation)
- Deploy preferences for Office for Mac
- Account Privacy Settings

Use preferences to manage privacy controls for Office on iOS devices

8/25/2021 • 4 minutes to read • Edit Online

NOTE

For a list of Office products covered by this privacy information, see Privacy controls available for Office products.

There are new preference settings for Office on iOS devices that allow you to control settings related to the following:

- ***Diagnostic data*** that is collected and sent to Microsoft about Office client software being used.
- ***Connected experiences*** that use cloud-based functionality to provide enhanced Office features to you and your users.

For more information about diagnostic data and connected experiences, see Overview of privacy controls.

NOTE

For information about similar settings for Office on computers running macOS, see Use preferences to manage privacy controls for Office for Mac

Setting device preferences

These new preference settings can also be set at the device level by a Mobile Device Management (MDM) server when the Office application is installed. Many MDM servers allow IT administrators to add an optional configuration dictionary when the server sends the `InstallApplication` MDM command to an iOS device. Refer to your MDM server documentation for more details.

The dictionary is represented as a set of key/value pairs in XML format. For example:

```
<dict>
 <key>DiagnosticDataTypePreference</key>
 <string>BasicDiagnosticData</string>
</dict>
```

Once sent to the device, the configuration dictionary will reside under the `com.apple.managed.configuration` key, where it will be read when the Office application is launched.

NOTE

You can also use the Office cloud policy service and these 4 policy settings:

- Configure the level of client software diagnostic data sent by Office to Microsoft
- Allow the use of connected experiences in Office that analyze content
- Allow the use of connected experiences in Office that download online content
- Allow the use of additional optional connected experiences in Office

Privacy settings for Outlook for iOS and OneDrive for iOS can only be configured by using the Office cloud policy service.

For more information on using the Office cloud policy service, see Overview of the Office cloud policy service.

Preference setting for diagnostic data

Diagnostic data is used to keep Office secure and up-to-date, detect, diagnose and remediate problems, and also make product improvements. For more information, see Diagnostic data sent from Microsoft 365 Apps for enterprise to Microsoft.

CATEGORY	DETAILS
Key	`DiagnosticDataTypePreference`
Data Type	String
Possible values	`BasicDiagnosticData` *(this value sets the level to Required)* `FullDiagnosticData` *(this value sets the level to Optional)* `ZeroDiagnosticData` *(this value sets the level to Neither)*

If you don't set this preference, both required and optional diagnostic data are sent to Microsoft if users with an Office 365 (or Microsoft 365) subscription are signed in with a work or school account. Also, these users can't change the level of diagnostic data regardless of how you set this preference.

For other users, such as home users with an Office 365 (or Microsoft 365) subscription, only required diagnostic data is sent, unless the user chooses to also send optional diagnostic data by going to **Settings** > **Privacy Settings**.

Preference setting for connected experiences that analyze your content

Connected experiences that analyze your content are experiences that use your Office content to provide you with design recommendations, editing suggestions, data insights, and similar features. For example, Design Ideas in PowerPoint. For a list of these connected experiences, see Connected experiences in Office.

CATEGORY	DETAILS
Key	`OfficeExperiencesAnalyzingContentPreference`
Data Type	Boolean
Possible values	`TRUE` *(enabled)* `FALSE` *(disabled)*

If you don't set this preference, connected experiences that analyze content are available to users.

If the user has an Office 365 (or Microsoft 365) subscription and is signed in with a work or school account, then the user can't turn off connected experiences that analyze content.

For other users, such as home users with an Office 365 (or Microsoft 365) subscription, the user can choose to turn off connected experiences that analyze content by going to **Settings** > **Privacy Settings**.

Preference setting for connected experiences that download online content

Connected experiences that download online content are experiences that allow you to search and download online content including templates, images, videos, and reference materials to enhance your documents. For example, Office templates or inserting an online icon. For a list of these connected experiences, see Connected experiences in Office.

CATEGORY	DETAILS
Key	`OfficeExperiencesDownloadingContentPreference`
Data Type	Boolean
Possible values	`TRUE` *(enabled)* `FALSE` *(disabled)*

If you don't set this preference, connected experiences that download online content are available to users.

If the user has an Office 365 (or Microsoft 365) subscription and is signed in with a work or school account, then the user can't turn off connected experiences that download online content.

For other users, such as home users with an Office 365 (or Microsoft 365) subscription, the user can choose to turn off connected experiences that download online content by going to **Settings** > **Privacy Settings**.

Preference setting for optional connected experiences

In addition to the connected experiences mentioned above, there are some optional connected experiences that you may choose to allow your users to access with their organization account, which is sometimes referred to as a work or school account. For example, Office add-ins that are downloaded through the Office Store to your device. For more examples, see Overview of optional connected experiences in Office.

CATEGORY	DETAILS
Key	`OptionalConnectedExperiencesPreference`
Data Type	Boolean
Possible values	`TRUE` *(enabled)* `FALSE` *(disabled)*

If you don't set this preference, optional connected experiences are available to users with an Office 365 (or Microsoft 365) subscription that are signed in with a work or school account. Unless you have set this preference to FALSE, these users can choose to turn off optional connected experiences by going to **Settings** > **Privacy Settings**.

For other users, such as home users with an Office 365 (or Microsoft 365) subscription, there isn't an option to turn off optional connected experiences.

Use policy settings to manage privacy controls for Office on Android devices

8/25/2021 • 2 minutes to read • Edit Online

> **NOTE**
> For a list of Office products covered by this privacy information, see Privacy controls available for Office products.

There are policy settings for Office on Android devices that allow you to control settings related to the following:

- ***Diagnostic data*** that is collected and sent to Microsoft about Office client software being used.
- ***Connected experiences*** that use cloud-based functionality to provide enhanced Office features to you and your users.

For more information about diagnostic data and connected experiences, see Overview of privacy controls.

Policy settings available for Office on Android devices

The following table lists which policy settings are available for Office on Android devices and a link to additional information about each policy setting.

> **NOTE**
> - The additional information provided covers the policy settings for Office on devices running Windows. But the same information applies to Office on Android devices because they are the same policy settings.
> - The *Allow the use of connected experiences in Office* policy setting that's available for Office on devices running Windows does not apply to Office on Android devices.

NAME OF POLICY SETTING	ADDITIONAL INFORMATION
Configure the level of client software diagnostic data sent by Office to Microsoft	Policy setting for diagnostic data
Allow the use of connected experiences in Office that analyze content	Policy setting for connected experiences that analyze your content
Allow the use of connected experiences in Office that download online content	Policy setting for connected experiences that download online content
Allow the use of additional optional connected experiences in Office	Policy setting for optional connected experiences

Use Office cloud policy service to apply policy settings

To apply any of these 4 policy settings for Office on Android devices, you need to use the Office cloud policy service. For more information on using the Office cloud policy service, see Overview of the Office cloud policy service.

NOTE

If you previously used Office cloud policy service to configure these policy settings for Office on devices running Windows, those same settings will apply to Office on Android devices. For that to happen, you just need to sign in to the Office cloud policy service and the service will apply the settings automatically to Office on Android devices.

Use policy settings to manage privacy controls for Office for the web applications

8/25/2021 • 3 minutes to read • Edit Online

NOTE

For a list of Office products covered by this privacy information, see Privacy controls available for Office products.

As an administrator for your organization, you can control whether your users have the choice to use optional connected experiences when they use Office for the web applications, such as Word for the web or PowerPoint for the web. This choice is available to your users only if they're signed in with their work or school account when they use Office for the web applications. To control whether your users have the choice to use optional connected experiences, you use the *Allow the use of additional optional connected experiences in Office* policy setting.

Overview of optional connected experiences

Optional connected experiences are cloud-backed services that are available to your users when they're using Office. Examples of optional connected experiences include creating a map chart in Excel or inserting an online picture into your Word document, both of which rely on services provided by Microsoft Bing. The use of these cloud-backed services is optional.

Optional connected experiences are not covered by your organization's commercial agreement with Microsoft. Instead, optional connected experiences are offered by Microsoft directly to your users and are governed by the Microsoft Services Agreement. In some cases, third-party content or functionality are provided through these optional connected experiences and other terms may also apply.

Some optional connected experiences might not be available in Office for the web applications, but are available when using other versions of Office, such as the desktop version on a device running Windows.

For more information, see Overview of optional connected experiences in Office.

Configure the policy setting by using the Office cloud policy service

You can use the *Allow the use of additional optional connected experiences in Office* policy setting to control whether your users have the choice to use optional connected experiences. To configure this policy setting for Office for the web applications, you need to use the Office cloud policy service.

If you don't configure this policy setting, the choice to use optional connected experiences will be available to your users. If you disable this policy setting, your users won't be able to use any of the optional connected experiences.

For Office for the web applications, the policy setting applies to when your users are working on Office documents that are saved to web-based storage from Microsoft, such as OneDrive for Business or SharePoint Online.

Because you're using the Office cloud policy service, this policy setting also applies to when your users are using Office on Windows, Mac, iOS, or Android devices. You can't configure this policy setting just for when your users are using Office for the web applications. But, you can create a policy configuration that includes this policy setting and have that policy configuration only apply to users that access documents anonymously using Office

for the web applications.

If you choose to make optional connected experiences available to your users, your users will be shown a privacy notification the first time they use an Office for the web app. This notification lets your users know that you've given them the option to use these optional connected experiences. The notification also informs your users that the optional connected experiences are provided under the Microsoft Services Agreement. Because this notification is important information for your users, this notification must be shown and can't be turned off, hidden, or accepted on behalf of your users.

Users can choose to turn off optional connected experiences

If you choose to make optional connected experiences available to your users, your users can go to their account privacy settings and choose to turn off their access to optional connected experiences. This choice is available in the account privacy settings only if your users are signed in with their work or school account. There is no way that you, as the admin, can prevent individual users in your organization from turning off their access to optional connected experience in their account privacy settings if you've given your users the choice to use optional connected experiences.

Related articles

- Overview of privacy controls for Microsoft 365 Apps for enterprise
- Use policy settings to manage privacy controls for Microsoft 365 Apps for enterprise
- Use preferences to manage privacy controls for Office for Mac
- Use preferences to manage privacy controls for Office on iOS devices
- Use policy settings to manage privacy controls for Office on Android devices

Required diagnostic data for Office

8/25/2021 • 484 minutes to read • Edit Online

> **NOTE**
> For a list of Office products covered by this privacy information, see Privacy controls available for Office products.

Diagnostic data is used to keep Office secure and up-to-date, detect, diagnose and fix problems, and also make product improvements. This data does not include a user's name or email address, the content of the user's files, or information about apps unrelated to Office.

This diagnostic data is collected and sent to Microsoft about Office client software running on the user's device. Some diagnostic data is required, while some diagnostic data is optional. We give you the ability to choose whether to send us required or optional diagnostic data through the use of privacy controls, such as policy settings for organizations. You can see the diagnostic data being sent to us by using the Diagnostic Data Viewer.

Required diagnostic data is the minimum data necessary to help keep Office secure, up-to-date, and performing as expected on the device it's installed on.

Required diagnostic data helps to identify problems with Office that may be related to a device or software configuration. For example, it can help determine if an Office feature crashes more frequently on a particular operating system version, with newly introduced features, or when certain Office features are disabled. Required diagnostic data helps us detect, diagnose, and fix these problems more quickly so the impact to users or organizations is reduced.

For more information about diagnostic data, see the following articles:

- Optional diagnostic data for Office
- Using the Diagnostic Data Viewer with Office

If you're the admin for your organization, you might also be interested in the following articles:

- Overview of privacy controls for Microsoft 365 Apps for enterprise
- Use policy settings to manage privacy controls for Microsoft 365 Apps for enterprise
- Use preferences to manage privacy controls for Office for Mac
- Use preferences to manage privacy controls for Office on iOS devices
- Use policy settings to manage privacy controls for Office on Android devices

> **NOTE**
> For information about required diagnostic data for Microsoft Teams, see the following articles:
>
> - Required desktop diagnostic data for Microsoft Teams
> - Required mobile diagnostic data for Microsoft Teams

Categories, data subtypes, events, and data fields for required diagnostic data

Required diagnostic data is organized into categories and data subtypes. Within each data subtype are events, which contain data fields that are specific to that event.

The following table provides a list of the categories for required diagnostic data. The data subtypes within each category are listed, along with a description of the focus for that data subtype. There are links to each data subtype section where you'll find the following information:

- A list of events in that data subtype
- A description of each event
- A list of data fields in each event
- A description of each data field

CATEGORY	DATA SUBTYPE	DESCRIPTION
Software setup and inventory	Office setup and inventory	Installed product and version and the installation status.
	Office add-in configuration	Software add-ins and their settings.
	Security	Document, feature, and add-in error conditions that may compromise security, including product update readiness.
Product and service usage	Application feature success	Success of application functionality. Limited to opening and closing of the application and documents, file editing, and file sharing (collaboration).
	Application status and boot	Determination if specific feature events have occurred, such as start or stop, and if feature is running.
	Office accessibility configuration	Office accessibility features
	Privacy	Office privacy settings
Product and service performance	Unexpected application exit (crash)	Unexpected application exits and the state of the application when that happens.
	Application feature performance	Poor response time or performance for scenarios, such as application start up or opening a file.
	Application activity error	Errors in functionality of a feature or user experience.
Device connectivity and configuration	Device connectivity and configuration	Network connection state and device settings, such as memory.

NOTE

- Categories are shown in the Diagnostic Data Viewer, but data subtypes are not shown.
- A data field marked *Obsolete* has been or will soon be removed from required diagnostic data. Some of these data fields are duplicates that arose as diagnostic data was modernized and were used to ensure no service disruption to live diagnostic monitoring reports.

Categories and data fields that are common for all events

There is some information about events that is common to all events, regardless of category or data subtype. This common information, which is sometimes referred to as *data contracts*, is organized into categories. Each category contains fields, and these fields are the metadata and properties of an individual event. You can view this information by using the Diagnostic Data Viewer.

The categories of information that is collected about events can be divided into two groups:

- Information common to all events
- Information that specifically supports diagnostic data collection

Information common to all events

Information common to all events is collected in the following categories.

App

Information about the application. All fields are constant for all sessions of a given application version.

This category contains the following fields:

- **Branch** - The branch that the given build came from. Allows us to determine what type of branch a given build came from so that we can correctly target fixes.
- **InstallType** - An enumerator that identifies how the user installed the application. Allows us to determine if specific install mechanisms are creating issues that are not seen in other installation mechanisms.
- **Name** - The name of the application that is providing the data. Allows us to identify which application is showing an issue so we know how to address it.
- **Platform** - The broad classification of the platform on which the app is running. Allows us to identify on which platforms an issue may be occurring so that we can correctly prioritize the issue.
- **Version** - The version of the application. Allows us to identify which versions of the product are showing an issue so that we can correctly prioritize it.

Client

Identifier related to an Office instance on a device. Constant for all sessions of all apps of a given installation version for multi-app suites, or constant for all sessions of a given application version.

This category contains the following fields:

- **Id** - Unique identifier assigned to a client at install time of Office. Allows us to identify whether issues are impacting a select set of installs and how many users are impacted.

Consent

Information regarding the users consent for diagnostic data and connected experiences.

This category contains the following fields:

- **ControllerConnectedServicesSourceLocation** – Indicates how the user's choice for optional connected experiences was made
- **ControllerConnectedServicesState** – Indicates whether the user has access to optional connected experiences
- **ControllerConnectedServicesStateConsentTime** – Indicates when the user chose the status of optional connected experiences. The date will appear as either a human readable date or as a machine encoded date that looks like a large number.
- **DiagnosticConsentConsentTime** – Indicates when the user provided the consent for diagnostic data. The date will appear as either a human readable date or as a machine encoded date that looks like a large number.

- **DiagnosticConsentLevel** – Indicates what level of diagnostic data consent the user has given
- **DiagnosticConsentLevelSourceLocation** – Indicates how the user had provided the consent for diagnostic data
- **DownloadContentSourceLocation** – Indicates how the user made the choice to enable or disable connected experiences that that download online content
- **DownloadContentState** – Indicates whether the user has chosen to enable or disable connected experiences that download online content
- **DownloadContentStateConsentTime** – Indicates when the user made the choice to enable or disable connected experiences that download online content. The date will appear as either a human readable date or as a machine encoded date that looks like a large number.
- **ServiceConnectionState** – Indicates whether the user has chosen to use or not use all connected experiences
- **ServiceConnectionStateConsentTime** – Indicates when the user chose whether to use all connected experiences. The date will appear as either a human readable date or as a machine encoded date that looks like a large number.
- **ServiceConnectionStateSourceLocation** – Indicates how the user provided the choice whether to use all connected experiences
- **UserCategoryValue** – Identified the type of user who made the consent. One of MSAUser, AADUser or LocalDeviceUser
- **UserContentDependentSourceLocation** – Indicates how the user's choice to enable or disable was made for connected experiences that analyze content
- **UserContentDependentState** – Indicates whether the user has chosen to enable or disable connected experiences that analyze content
- **UserContentDependentStateConsentTime** – Indicates when the user chose to enable or disable connected experiences that analyze content was made. The date will appear as either a human readable date or as a machine encoded date that looks like a large number.

Device

Information about the operating system and build.

This category contains the following fields:

- **Model** - string containing the physical model for the device running the app. iOS only. For example, iPhone13,3 or iPad11,6.
- **OsBuild** - The build number of the operating system installed on the device. Allows us to identify whether issues are impacting individual service packs or versions of a given operating system differently than others so we can prioritize issues.
- **OsVersion** - The major version of the operating system installed on the device. Allows us to determine if issues are impacting a particular operating system version more than other so we can prioritize issues.

Legacy

Provides an App ID and OS version for compatibility with existing legacy collection practices.

This category contains the following fields:

- **AppId** - An enumerator value representing which application is sending the data. Allows us to identify which application is showing an issue so we know how to address it.

- **OsEnv** - An enumerator indicating which operating system the session is running on. Allows us to identify which operating system an issue happening on so we can prioritize issues.

Release

Information related to the release channel. All fields are constant for all sessions of all apps of a given installation version. Identifies a group of devices all in one phase of a product release cycle.

This category contains the following fields:

- **Audience** - Identifies a sub-audience of a given audience group. Allows us to track subsets of audience groups to evaluate prevalence and prioritization of issues.
- **AudienceGroup** - Identifies the ring where data is coming from. Allows us to roll out features in a staged fashion and identify potential issues before they reach most users.
- **Channel** - The channel that the product is being released through. Allows us to identify if an issue is impacting one of our rollout channels differently than others.
- **Fork** - Identifies the fork of the product. Allows a mechanism to aggregate data across a set of build numbers to identify issues related to a given release.

Session

Information about the process session. All fields are constant for this session.

This category contains the following fields:

- **ABConfigs** - Identifies the set of flights that are running in a given session. Allows us to identify which individual flights are running on a session so that we can determine if a flight is the source of an issue impacting users.
- **EcsETag** - An indicator from the flighting system that represents the flights sent to the machine. Allows us to identify what flights might be impacting a given session.
- **Flags** - Bitmask tracking flags applicable to an entire session, currently primarily focused on sampling and diagnostic data options. Allows us to control how a given session behaves in relation to the diagnostic data that the session generates.
- **HostAppName** - Identifies the host app name that launches a sub-app. Apps like Office Mobile (Android) can launch Word, Excel, and PowerPoint sub-apps. For such sub-apps, the host app is OfficeMobile
- **HostSessionId** - Uniquely identifies the host app session for a sub-app
- **Id** - Uniquely identifies a given session of data. Allows us to identify the impact of issues by evaluating the number of sessions that are impacted and if there are common features of those sessions.
- **ImpressionId** - Identifies the set of flights that are running in a given session. Allows us to identify which individual flights are running on a session so that we can determine if a flight is the source of an issue impacting users.
- **MeasuresEnabled** - Flag indicating if the current sessions data is sampled or not. Allows us to determine how to statistically evaluate the data that is gathered from the given session.
- **SamplingClientIdValue** - The ID of the client used to determine if it is part of sampling. Allows us to determine why an individual session was included or excluded from sampling.
- **SubAppName** - For Office Mobile app, this field represents the underlying application being used to open a document. For example, if you open a Word document in Office app, this field will report the value of “Word”.

- **VirtualizationType** - Type of virtualization if Office is running in one. The possible values are:
  - 0 = None
  - 1 = Windows Virtual Desktop
  - 2 = Windows Defender Application Guard
  - 3 = Windows Core OS

User

Provides tenant information for commercial software SKUs.

This category contains the following fields:

- **PrimaryIdentityHash** – A pseudonymous identifier that represents the current user.
- **PrimaryIdentitySpace** – The type of identity contained in the PrimaryIdentityHash. One of MASCID, OrgIdCID or UserObjectId.
- **TenantGroup** - The type of the tenant that the subscription belongs to. Allows us to classify issues and identify whether a problem is widespread or isolated to a set of users.
- **TenantId** - The tenant that a user's subscription is tied to. Allows us to classify issues and identify whether a problem is widespread or isolated to a set of users or a specific tenant.

Information that specifically supports diagnostic data collection

Information that specifically supports diagnostic data collection is collected in the following categories.

Activity

Information to understand the success of the collection event itself.

This category contains the following fields:

- **AggMode** - Tells the system how to aggregate activity results. Allows us to reduce the amount of information uploaded from a user's machine by aggregating activity results into a single event that gets sent periodically.
- **Count** - The number of times the activity happened if the count is from an aggregated event. Allows us to determine how often an activity succeeded or failed based on the aggregation mode of the activity.
- **CV** - A value that identifies the relationship between activities and sub-activities. Allows us to rebuild the relationship between nested activities.
- **Duration** - The length of time the activity took to execute. Allows us to identify performance issues that are negatively impacting the user's experience.
- **Result.Code** - An application defined code to identify a given result. Allows us to determine more specific details of a given failure such as a failure code that can be used to classify and fix issues.
- **Result.Tag** - An integer tag that identifies the location in code where the result was generated. Allows us to distinctly identify the location in code where a result was generated which enables classification of failures.
- **Result.Type** - The type of the result code. Identifies what type of result code was sent so that the value can be correctly interpreted.
- **Success** - A flag indicating if the activity succeeded or failed. Allows us to determine if actions the user takes in the product are succeeding or failing. This allows us to identify issues that are impacting the user.

Application

Information about the installation of the application from which the events are being gathered.

This category contains the following fields:

- **Architecture** - The architecture of the application. Let's us classify errors that might be specific to an architecture of the application.
- **Click2RunPackageVersion** - The version number of the Click-To-Run package that installed the app. Allows us to identify which version of the installer was used to install Office so we can identify setup-related issues.
- **DistributionChannel** - The channel where the app was deployed. Allows us to partition incoming data so we can determine if issues are impacting audiences.
- **InstallMethod** - Whether the current build of Office was upgraded from an older build, rolled back to an older build, or a fresh install.
- **IsClickToRunInstall** - Flag indicating if install was a click to run install. Allows us to identify issues that might be specific to the Click-To-Run install mechanism.
- **IsDebug** - Flag indicating if the Office build is a Debug build. Allows us to identify if issues are coming from Debug builds, which may behave differently.
- **IsInstalledOnExternalStorage** - Flag indicating if Office was installed on an external storage device. Let's us determine if issues can be traced to an external storage location.
- **IsOEMInstalled** - Flag indicating if Office was installed by an original equipment manufacturer (OEM). Let's us determine if the application was installed by an OEM, which can help us classify and identify issues.
- **PreviousVersion** - The version of Office that was previously installed on the machine. Allows us to roll back to a previous version if the current one has an issue.
- **ProcessFileName** - The name of the application filename. Allows us to identify the name of the executable that is generating the data as there may be several different process filenames reporting as the same app name.

Client

Information about the Office client.

This category contains the following fields:

- **FirstRunTime** - The first time the client was run. Allows us to understand how long the client has had Office installed.

Device

Information about device configuration and capabilities.

This category contains the following fields:

- **DigitizerInfo** - Information on the digitizer used by the machine. Allows us to classify data based on device pivot.
- **FormFactor** - Identifies what form factor the device sending the info is. Allows us to classify data based on device pivot.
- **FormFactorFamily** - Identifies what form factor the device sending the info is. Allows us to classify data based on device pivot.
- **HorizontalResolution** - The horizontal resolution of the devices screen. Allows us to classify data based on device pivot.
- **Id** - A unique identifier for the device. Allows us to identify the distribution of issues across a set of

devices.

- **IsEDPPolicyEnabled** - Flag to indicate if enhanced data protection is enabled on the machine. Allows us to classify data based on device pivot.
- **IsTerminalServer** - Flag to determine if the machine is a terminal server. Allows us to classify data based on device pivot.
- **Manufacturer** - The manufacturer of the device. Allows us to classify data based on device pivot.
- **Model** - The model of the device. Allows us to classify data based on device pivot.
- **MotherboardUUIDHash** - A hash of a unique identifier for the motherboard. Allows us to classify data based on device pivot.
- **Name** - The name of the device. Allows us to classify data based on device pivot.
- **NetworkCost** - Indicates network cost or type, such as metered or metered above cap.
- **NetworkCountry** - The country code of the sender, based on the unscrubbed client IP address.
- **NumProcPhysCores** - The number of physical cores on the machine. Allows us to classify data based on device pivot.
- **OsLocale** - The locale of the operating system that is running. Allows us to classify data based on device pivot.
- **ProcessorArchitecture** - The architecture of the processor. Allows us to classify data based on device pivot.
- **ProcessorCount** - The number of processors on the machine. Allows us to classify data based on device pivot.
- **ProcSpeedMHz** - The speed of the processor. Allows us to classify data based on device pivot.
- **RamMB** - The amount of memory the device has. Allows us to classify data based on device pivot.
- **ScreenDepth** - Allows us to classify data based on device pivot.
- **ScreenDPI** - The DPI value of the screen. Allows us to classify data based on device pivot.
- **SusClientId** - The Windows Update ID of the device Office is run on.
- **SystemVolumeFreeSpaceMB** - The amount of free space on the system volume. Allows us to classify data based on device pivot.
- **SystemVolumeSizeMB** - The size of the system volume on the machine. Allows us to classify data based on device pivot.
- **VerticalResolution** - The vertical resolution of the devices screen. Allows us to classify data based on device pivot.
- **WindowErrorReportingMachineId** - A unique machine identifier provided by Windows error reporting. Allows us to classify data based on device pivot.
- **WindowSqmMachineId** - A unique identifier for the machine provided by Windows SQM. Allows us to classify data based on device pivot.

Event

Event-specific information, including its unique identifier in the session.

This category contains the following fields:

- **Contract** - A list of any contracts that the event is implementing. Allows us to evaluate what data is part of the individual event so that we can process it effectively.
- **CV** - A value that allows us to identify events that are related to one another. Used for diagnostics to allow us to identify patterns of related behavior or related events.
- **Flags** - Information used to alter how a given event responds. Used to manage how a given event is treated for the purposes of uploading the data to Microsoft.
- **Id** - A unique identifier for the event. Allows us to uniquely identify the events that are being received.
- **IsExportable** - A field to denote if this event needs further processing by export pipeline.
- **Level** - denotes the type of event.
- **Name** - The name of the event. Allows us to identify the event that was being sent from the client.
- **Rule** - An identifier of the rule that generated the data if it was generated by a rule. Allows us to identify the source of a piece of data so that we can validate and manage that events parameters
- **RuleId** - The identifier of the rule that generated the data if it was generated by a rule. Allows us to identify the source of a piece of data so that we can validate and manage that events parameters.
- **RuleInterfaces** - Any interfaces that are implemented by the specific rule. Allows us to classify and import the data based on its structure, which simplifies data processing.
- **RuleVersion** - The identifier of the rule that generated the data if it was generated by a rule. Allows us to identify the source of a piece of data so that we can validate and manage that events parameters.
- **SampleRate** - An indication of what percentage of users are sending this piece of data. This allows us to do statistical analysis of the data and for very common data points not require that to be sent by all users.
- **SchemaVersion** - The version of the schema used to generate diagnostic data. Required to manage data being sent from the client. This allows changes over time in what data is being sent from each client.
- **Sequence** - A counter that identifies the order that an event was generated on the client. Allows the data being received to be ordered so that we can identify what steps may have led to an issue that is impacting clients.
- **Source** - The source pipeline that was used to upload the data. Required to monitor each of our upload pipelines for overall health and to help identify issues with the upload pipeline. This allows us to monitor individual upload pipelines to make sure they remain compliant.
- **Time** - The time that the event was generated on the client. Allows us to synchronize and validate the order of events generated on the client and establish performance metrics for user instructions.

Host

Information about an application that hosts an embedded application

This category contains the following fields:

- **Id** - The unique identifier attributed to the host application by the embedded application.
- **SessionId** - The globally unique identifier for the host's session.
- **Version** - The version identifier of the host's primary executable.

Legacy

Information needed for legacy system compatibility.

This category contains the following fields:

- **OsBuild** - The specific build number of the operating system. Allows us to determine which version of the operating system the diagnostic data is coming from in order to prioritize issues.
- **OsBuildRevision** - The revision number of the build of the operating system. Allows us to determine which version of the operating system the diagnostic data is coming from in order to prioritize issues.
- **OsMinorVersion** - The minor version of the operating system. Allows us to determine which version of the operating system the diagnostic data is coming from in order to prioritize issues.
- **OsVersionString** - A unified string representing the operating system build number. Allows us to determine which version of the operating system the diagnostic data is coming from in order to prioritize issues.

Session

Information about the process session.

This category contains the following fields:

- **ABConfigsDelta** - Tracks the difference between the current ABConfigs data and the previous value. Allows us to track what new flights are on the machine to help identify if a new flight is responsible for an issue.
- **CollectibleClassification** - The classes of information the session can collect. Allows filtering of sessions based on the data they would have.
- **DisableTelemetry** - Flag indicating if the DisableTelemetry key is set. Allows us to know if a session was not reporting diagnostic data other than EssentialServiceMetadata.
- **SamplingClientIdValue** - The value of the key used to determine sampling. Allows us to determine why a session was sampled or not.
- **SamplingDeviceIdValue** - The value of the key used to determine sampling. Allows us to determine why a session was sampled or not.
- **SamplingKey** - The key used to determine whether the session is sampled or not. Allows us to understand how individual sessions are making their choice of whether they are sampled or not.
- **SamplingMethod** - The method used to determine sampling policy. Allows us to understand what data is coming from a session.
- **SamplingSessionKValue** - Advanced sampling metadata. Used to help evaluate statistical meaning of data that is received.
- **SamplingSessionNValue** - Advanced sampling metadata. Used to help evaluate statistical meaning of data that is received.
- **Sequence** - A unique numeric identifier for the session. Allows the ordering of sessions for analysis of the issues might have occurred.
- **Start** - The boot time of the process session. Allows us to establish when the session started.
- **TelemetryPermissionLevel** - Value indicating what level of diagnostic data the user has opted into. Allows us to understand what level of diagnostic data to expect from a session.
- **TimeZoneBiasInMinutes** - The difference in minutes between UTC and the local time. Allows normalization of UTC times back to the local time.

Data fields that are common for OneNote events

The following data fields are common for all events for OneNote on Mac, iOS, and Android.

> **NOTE**
> When using the Diagnostic Data Viewer, events for OneNote on Mac, iOS, and Android will appear to have a name of Activity, ReportData, or Unexpected. To find the actual event name, select the event, and then look at the EventName field.

- **Activity_ActivityType** - Indicates the type of this activity event. An activity can be a normal activity or a high value activity.
- **Activity_AggMode** - Tells the system how to aggregate activity results. Allows us to reduce the amount of information uploaded from a user's machine by aggregating activity results into a single event that gets sent periodically.
- **Activity_Count** - The number of times the activity happened if the count is from an aggregated event. Allows us to determine how often an activity succeeded or failed based on the aggregation mode of the activity.
- **Activity_CV** - A value that identifies the relationship between activities and sub-activities. Allows us to rebuild the relationship between nested activities.
- **Activity_DetachedDurationInMicroseconds** - The length of time an activity is idle and not doing any real work, but the time is still count towards the total activity's time.
- **Activity_DurationInMicroseconds** - The length of time the activity took to execute. Allows us to identify performance issues that are negatively impacting the user's experience.
- **Activity_Expiration** - A date in numerical format indicates when this event will be stop sending from clients
- **Activity_FailCount** - The number of times this activity has failed
- **Activity_Name** - A short name of an event. Allows us to identify the event that was being sent from the client.
- **Activity_Namespace** - A namespace of an event. Allows us to group the event into groups.
- **Activity_Reason** - A string indicating the reason causing an activity to ends with a particular result.
- **Activity_Result** - A flag indicating if the activity succeeded, failed, or unexpectedly failed. Allows us to determine if actions the user takes in the product are succeeding or failing. This allows us to identify issues that are impacting the user.
- **Activity_State** - A flag indicates whether an event is a start of a user activity or an end of a user activity.
- **Activity_SucceedCount** - The number of times this activity succeeded.
- **ErrorCode** - Indicates an error code if available.
- **ErrorCode2** - Indicates a second error code if available.
- **ErrorCode3** - Indicates a third error code if available.
- **ErrorTag** - Indicates the tag associated in code of an error if available.
- **ErrorType** - Indicates the type of an error if available.
- **EventName** - A unique name of a OneNote's event. OneNote events use this custom field to specify a unique name due to an engineering limitation in the past.
- **ExpFeatures** - Indicates whether a user has turn-on an experimental-feature switch in OneNote app or not.

- **ExpirationDate** - A date in numerical format indicates when this event will be stop sending from clients
- **IsConsumer** - Indicates whether a user is consumer or not
- **IsEdu** - Indicates whether a user is a user in education tenant or not
- **IsIW** - Indicates whether a user is an enterprise user or not
- **IsMsftInternal** - Indicates whether a user is a Microsoft employee or not
- **IsPremiumUser** - Indicates whether a user has premium license or not
- **Namespace** - A namespace of the event. Allows us to group the event into groups.
- **Release_AppStore** - A flag indicates whether a build is coming from an app store or not.
- **Release_Audience** - Identifies a sub-audience of a given audience group. Allows us to track subsets of audience groups to evaluate prevalence and prioritization of issues.
- **Release_AudienceGroup** - Identifies the ring where data is coming from. Allows us to roll out features in a staged fashion and identify potential issues before they reach most users.
- **Release_Channel** - The channel that the product is being released through. Allows us to identify if an issue is impacting one of our rollout channels differently than others.
- **RunningMode** - Indicates how the app is launched either by user or by system process.
- **SchemaVersion** - Indicates a current telemetry schema version in OneNote's telemetry pipeline.
- **Session_EcsETag** - An indicator from the flighting system that represents the flights sent to the machine. Allows us to identify what flights might be impacting a given session.
- **Session_ImpressionId** - Identifies the set of flights that are running in a given session. Allows us to identify which individual flights are running on a session so that we can determine if a flight is the source of an issue impacting users.
- **SessionCorrelationId** - The globally unique identifier for the host's session.
- **SH_ErrorCode** - Indicates an error code if available when an activity fails.
- **Tag** - An integer tag that identifies the location in code where the telemetry event is generated.
- **UserInfo_IdType** - A string indicates the type of a user's account
- **UserInfo_OMSTenantId** - The tenant that a user's subscription is tied to. Allows us to classify issues and identify whether a problem is widespread or isolated to a set of users or a specific tenant.
- **UserInfo_OtherId** - A list of non-primary pseudonymous identifiers representing user's accounts.
- **UserInfo_OtherIdType** - A list of non-primary account types.

Data fields that are common for Outlook mobile events

Outlook mobile collects common fields for each of our events so that we can ensure the app is up-to-date, secure, and functioning as expected.

The following data fields are common for all events for Outlook for iOS and Android.

- **aad_tenant_id** - The tenant ID of the customer if available
- **account_cid** - A pseudonymous identifier that represents the current user
- **account_domain** - Domain name of the account

- **account_puid** - The globally unique user identifier for a consumer Microsoft account
- **account_type** - Tracks the account type such as Office 365, Google Cloud Cache, Outlook.com, etc.
- **action** - The event action name (such as archive, delete, etc.) so we can detect issues with specific actions taken
- **ad_id** - The unique advertising identifier
- **app_version** - Current version of the app installed to help us detect issues affecting certain app version
- **AppInfo.ETag** - A unique identifier for managing release of our features to help us detect issues affecting certain features being released
- **AppInfo.Language** - Currently language setting of the device to help us detect issues affecting certain languages
- **AppInfo.Version** - Current version of the app installed to help us detect issues affecting certain app versions
- **ci** - a pseudonymous app-specific device unique identifier
- **cid_type** - indicates what type of account it is, such as a commercial account or Outlook.com account.
- **cloud** - Where the mailbox resides for the account on this device to help detect issues specific to a specific mailbox cloud, like Office 365 or GCC.
- **customer_type** - Indicates the type of customer (consumer, commercial, third party, etc.) to help us detect issues affecting certain customer types
- **device_category** - Indicates what type of device it is (phone, tablet, etc.) to help us detect device category-specific issues
- **DeviceInfo.Id** - A unique device identifier to help us detect device-specific issues
- **DeviceInfo.Make** - The make of the device (for example, Apple, Samsung, etc.) to help us detect device make specific issues
- **DeviceInfo.Model** - The device model (for example, iPhone 6s) to help us detect device model specific issues
- **DeviceInfo.NetworkType** - The current device network being used (WiFi, cellular, etc.) to help us detect device network specific issues
- **DeviceInfo.OsBuild** - The current OS build of the device to help us detect issues affecting certain OS builds
- **DeviceInfo.OsName** - The name of the OS (for example, iOS) to help us detect issues affecting certain platforms
- **DeviceInfo.OsVersion** - The current OS version of the device to help us detect issues affecting certain OS versions
- **DeviceInfo.SDKUid** - The device unique identifier (similar to DeviceInfo.Id)
- **EventInfo.InitId** - ID used as part of sequencing for event ordering through our telemetry pipeline to help us detect the root cause of a pipeline issue
- **EventInfo.SdkVersion** - The SDK version we are using to send our telemetry to help us detect the root cause of a pipeline issue
- **EventInfo.Sequence** - The sequence is for event ordering through our telemetry pipeline to help us

detect the root cause of a pipeline issue

- **EventInfo.Source** - Tells us what part of the code sent the event to help us detect the root cause of an issue
- **EventInfo.Time** - The time and date the event was emitted from the device so our systems can successfully manage events coming in
- **eventpriority** - The priority of the telemetry event relative to other events so our systems can successfully manage events coming in
- **first_launch_date** - The first time the app was launched so helps us understand when an issue first began
- **hashed_email** - A pseudonymous identifier that represents the current user email
- **is_first_session** - Tracks if this is the first session of the app for debugging purposes
- **origin** - The origination of an action. For example marking a message read can originate from message list or a new mail notification, this helps us detect issues based on action origination
- **PipelineInfo.AccountId** - A pseudonymous identifier that represents the current user
- **PipelineInfo.ClientCountry** - The current country of the device to detect country or region-specific issues and outages
- **PipelineInfo.ClientIp** - The IP address the device is connected to debug connection issues
- **PipelineInfo.IngestionTime** - Timestamp of when our telemetry ingestion happens for this event
- **sample_rate** - The percentage of devices that collect instances of the event. Helps calculate the original number of instances of the event.
- **Session.Id** - A unique identifier for the app session to help identify session-related issues
- **Session.ImpressionId** - A unique identifier for managing release of our features to ensure features are successfully released to all users and devices
- **ui_mode** - Is the user in light or dark mode, helps us triage UX bugs with dark mode
- **UserInfo.Language** - The user's language to help debug translation text issues
- **UserInfo.TimeZone** - The user's time zone to help debug calendar issues

In addition, the following fields are common for all events for Outlook for iOS.

- **DeviceInfo.NetworkProvider** - The network provider of the device (for example, Verizon)
- **gcc_restrictions_enabled** - Tells us if GCC restrictions have been applied to the app so we can ensure our GCC customers are using our app securely
- **multi_pane_mode** - Tells us if the user on the iPad is using their inbox with multiple panes turned on where they can see their folder list while triaging email. This is needed to help us detect issues specific to those using their inbox with multiple panes open.
- **multi_window_mode** – Tells us if the user on the iPad is using multiple windows to help us detect issues related to multi-window usage.
- **office_session_id** - A unique ID tracking the session for connected Office services to help detect issues specific an Office service integration in Outlook like Word
- **state** - Whether the app was active when this event was sent to help detect issues specific to active or

inactive app states

In addition, the following fields are common for all events for Outlook for Android.

- **aad_id** - a pseudonymous Azure Active Directory identifier
- **DeviceInfo.NetworkCost** - Indication of devices current network cost, which reflects the status of WiFi/Cellular/Roaming to help detect issues specific to device network
- **is_app_in_duo_split_view_mode** - This will let us know that the app was in Duo split-screen mode. This property is set only for Duo (Android only) devices.
- **is_dex_mode_enabled** - Whether Samsung DeX mode is enabled to help detect issues specific to DeX mode with Samsung devices
- **is_preload_install** – Tells us if our app was pre-loaded on device (Android 11 or later devices)
- **is_sliding_drawer_enabled** - Whether Sliding Drawer interface is enabled to help detect issues caused by our sliding drawer interface
- **oem_preinstall** - Tells us if our app was pre-installed on the device
- **oem_preload_property** – Tells us if our app was pre-loaded as part of a specific agreement with the OEM
- **orientation** - Physical orientation of the screen (portrait/landscape) to help detect issues specific to device orientation
- **os_arch** - Operating System architecture for the device to help detect issues specific to device operation systems
- **process_bitness** - Process bitness (32 or 64 bit) for the application to help detect issues specific to device bitness
- **webview_kernel_version**: The Chromium kernel version of webview on the device to help us detect compatibility issues related to the version of webview.
- **webview_package_name**: The package name of webview on the device to help us detect compatibility issues related to the version of webview.
- **webview_package_version**: The package version of webview on the device to help us detect compatibility issues related to the version of webview.

Software setup and inventory data events

The following are the data subtypes in this category:

- Office setup and inventory
- Office add-in configuration
- Security

Office setup and inventory subtype

Installed product and version and the installation status.

add.sso.account

This will alert Microsoft to the success or failure of a user adding an account through single sign-on (SSO).

The following fields are collected:

- **account_type** – the type of the account added using the SSO.

- **action_origin** – from where this event was generated. (for example, values: sso_drawer, sso_add_account, sso_add_account_prompt, sso_settings, sso_oobe).
- **provider** - the identifier for the provider software package for the SSO.
- **state** – current state of the account, (example value: FAILED, PENDING, ADDED etc.)

install.referral

This event is triggered at the initial install of the app and records from where the user was referred (if available).

The following fields are collected:

- **install_referrer** - Product or experience from where the user was referred

Office.ClickToRun.UpdateStatus

Applicable to all win32 applications. Helps us understand the status of the update process of the Office suite (Success or failure with error details)

The following fields are collected:

- **build** - Currently installed Office version
- **channel** - The channel by which Office was distributed
- **errorCode** - Error code indicating the failure
- **errorMessage** - Additional error information
- **status** - Current status of the update
- **targetBuild** - Version Office is updating to

Office.Compliance.FileFormatBallotDisplayedOnFirstBoot

Indicates whether the Office File Format choice dialog box was shown to the user on first/second boot of Word, Excel, PowerPoint on Win32. Tracks whether the FileFormat Ballot dialog box is displayed - event is sent at the first/second boot of Word, Excel, or PPT Win32.

The following fields are collected.

- **CountryRegion** – The users' country region setting in Windows system
- **FileFormatBallotBoxAppIDBootedOnce** – In which app (Word, Excel, PPT) the file format ballot display logic was processed.
- **FileFormatBallotBoxDisplayedOnFirstBoot** – What is the display result for file format ballot (displayed/not displayed as unexpected/not displayed due to license/not displayed due to location).

Office.Compliance.FileFormatBallotOption

Tracks whether the FileFormat Ballot dialog box is displayed - event is sent at the first/second boot of Word, Excel, or PPT Win32. Indicates whether the Office File Format choice dialog box is displayed on first or second boot of Word, Excel, or PowerPoint on Win32.

The following fields are collected:

- **FileFormatBallotSelectedOption** – Identifies the file format option (OOXML/ODF) that was selected by the user through the file format ballot dialog box.

Office.CorrelationMetadata.UTCCorrelationMetadata

Collects Office metadata through UTC to compare with equivalent data collected through the Office telemetry pipeline to check correctness and completeness of data.

The following fields are collected:

- **abConfigs** - A list of feature IDs to identify which features are enabled on the client or empty when this data is not being collected.
- **abFlights** - "NoNL:NoFlights" when the feature flights aren't set. Otherwise "holdoutinfo=unknown".
- **AppSessionGuid** - An identifier of a particular application session starting at process creation time and persisting until process end. It is formatted as a standard 128-bit GUID but constructed of four parts. Those four parts in order are (1) 32-bit Process ID (2) 16-bit Session ID (3) 16 bit Boot ID (4) 64-bit Process creation time in UTC 100 ns
- **appVersionBuild** - The app build version number.
- **appVersionMajor** - The app major version number.
- **appVersionMinor** - The app minor version number.
- **appVersionRevision** - The app revision version number.
- **audienceGroup** - The release audience group name
- **audienceId** - The release audience name
- **channel** - The channel by which Office was distributed
- **deviceClass** - Device form factor from the OS
- **ecsETag** - An experiment identifier for the process
- **impressionId** - A guid indicating the current set of features.
- **languageTag** - The current Office UI IETF language tag
- **officeUserID** - Randomly generated guid for this Office install
- **osArchitecture** - Operating system architecture
- **osEnvironment** - An integer indicating the operating system (Windows, Android, iOS, Mac, etc.).
- **osVersionString** - Operating system version
- **sessionID** - Randomly generated guid to identify the app session
- **UTCReplace_AppSessionGuid** - Constant boolean value. Always true.

Office.OneNote.Android.App.OneNoteLaunchedNonActivated

[This event was previously named OneNote.App.OneNoteLaunchedNonActivated.]

Records information about activation state of the App. The data is monitored to ensure we identify spikes in activation issues. We also analyze the data to find areas of improvement.

The following fields are collected:

- **INSTALL_LOCATION** - Indicates if the app is pre-installed or is downloaded from Store

Office.OneNote.Android.ResetStatus

[This event was previously named OneNote.ResetStatus.]

The signal used to record any issues encountered when a user tries to reset the App. The telemetry is used to monitor, detect, and fix any issues caused during reset.

The following fields are collected:

- **Accounts** - Indicates the types if accounts used for signing-into the App

- **Generic String Type** - Returns if it is full reset of a notes_light_data reset
- **LaunchPoint** - The point from where Reset is initiated. Possible values: Sign Out Button, Sign out failure, Intune Triggered
- **Pass** - Indicates if the Reset was successful

Office.OneNote.Android.SignIn.SignInCompleted

[This event was previously named OneNote.SignIn.SignInCompleted.]

The critical signal used to ensure sign-in successful or not. The telemetry is collected to ensure critical regression detection for OneNote app and service health

The following fields are collected:

- **CompletionState** - Final state of sign in - Succeeded or failed. And failure cases
- **EntryPoint** - Indicates from where Sign-In was initiated
- **Hresult** - Error code
- **Provider Package ID** - In case of Auto sign in
- **Result** - Succeeded, Failed, Unknown, Canceled
- **ServerType** - Returns the type of the server offering the service
- **SignInMode** - Sign in or Sign up or Auto Sign-in or Sign up accelerated

Office.OneNote.Android.SignIn.SignInStarted

[This event was previously named OneNote.SignIn.SignInStarted.]

The signal used to indicate any issues encountered while using Message Bar. The telemetry is used to monitor, detect, and fix any issues caused during interaction with Message Bar

The following fields are collected:

- **EntryPoint** - Indicates from where Sign-In was initiated
- **Result** - Result of the sign-in flow
- **ServerType** - Returns the type of the server offering the service
- **SignInMode** - Sign in or Sign up or Auto Sign in or Sign up accelerated

Office.OneNote.FirstRun.FirstRun

The critical signal used to ensure new users can successfully launch and run OneNote for the first time. The telemetry is collected to ensure critical regression detection for OneNote app and service health. If users can't launch the app for the first time, this would trigger a high severity incident.

- **AfterOneDriveFrozenAccountError** - Indicates an error from OneDrive when an account is frozen.
- **Attempt** - The number of times that the first run experience needs to retry.
- **IsDefaultNotebookCreated** - Indicates whether OneNote has created a user's default notebook or not.
- **IsDelayedSignIn** - Indicates whether the first run is in delayed sign-in mode where a user is not required to signed-in.
- **IsMSA** - Indicates whether an account is Microsoft account or not.

Office.OneNote.FirstRun.FirstRunForMSA

The critical signal used to ensure new consumer users (Microsoft Account) can successfully launch and use OneNote for the first time.

Telemetry used to ensure critical regression detection for OneNote app and service health. If users can't launch the app for the first time, this would trigger a high severity incident.

The following fields are collected:

- **Attempt** - The number of times that the first run experience needs to retry.
- **Error A** - OneNote's error object indicates an error during the first run if any.
- **FAllowAddingGuide** - Indicates whether OneNote will allow creating a guide notebook or not.
- **FrozenOneDriveAccount** - Indicates whether a OneDrive account is frozen or not.
- **IsDefaultNotebookCreated** - Indicates whether OneNote has created a user's default notebook or not.
- **NoInternetConnection** - Indicates whether a device does not have internet connection.
- **ProvisioningFailure** - A OneNote error object indicating a provisioning error if any.
- **ProvisioningFinishedTime** - Indicates the end time when OneNote finishes provisioning a notebook during first run experience.
- **ProvisioningStartedTime** - Indicates the start time when OneNote starts provisioning a notebook during first run experience.
- **ShowSuggestedNotebooks** - Indicates whether OneNote will show a suggested notebook feature or not.

Office.OneNote.FirstRun.FirstRunForOrgId

The critical signal used to ensure new enterprise users (AAD/OrgID) can successfully launch and run OneNote for the first time. Telemetry used to ensure critical regression detection for OneNote app and service health. If users can't launch the app for the first time, this would trigger a high severity incident.

- **Attempt** - The number of times that the first run experience needs to retry.
- **Error** - A OneNote's error object indicates an error during the first run if any.
- **FAllowAddingGuide** - Indicates whether OneNote will allow creating a guide notebook or not.
- **IsDefaultNotebookCreated** - Indicates whether OneNote has created a user's default notebook or not.
- **ProvisioningFailure** - A OneNote's error object indicates a provisioning error if any.
- **ProvisioningFinishedTime** - Indicates the end time when OneNote finishes provisioning a notebook during first run experience.
- **ProvisioningStartedTime** - Indicates the start time when OneNote starts provisioning a notebook during first run experience.

Office.OneNote.FirstRun.MruReaderNoteBookEntries

The signal used to record any issues encountered when loading Notebooks during First Run. The telemetry is used to monitor, detect, and fix any issues in First run.

The following fields are collected:

- **OnPremNBCount** - Count of notebooks in On Prem Server
- **TotalNBCount** - Total count of notebooks associated with the User Account

Office.OneNote.System.AppLifeCycle.UserAccountInfo

This event is triggered for shared code and records values for type of accounts signed in via isEdu, isMsftInternal, isIW, isMSA. The data is collected the first time the queue becomes idle after launch. This marker

is used to track the types of accounts which have been signed in on the device. This will help us identify EDU users in OneNote.

The following fields are collected:

- **IsEdu** - Possible values - true/false
- **IsMSA** - Possible values - true/false
- **IsIW** - Possible values – true/false
- **IsMsftInternal** - Possible values – true/false

Office.TargetedMessaging.EnsureCached

Tracks if a package for Dynamic Canvas was downloaded. Considered a software configuration because the package must be successfully downloaded to enable the client to render the right experience. Is especially critical in consumer subscriptions where we use canvas to communicate to the user that the license has expired. Used to track metadata of a dynamic content package downloaded and cached by the product and results of operations performed on the package: download failures, unpacking failures, consistency checks failures, cache hits, package usages, download sources.

The following fields are collected:

- **Data_CacheFolderNotCreated** - Boolean flag indicating if cache folder creation succeeded
- **Data_CdnPath – source address of the package-**
- **Data_EnsureCached** - Boolean flag indicating if content package was cached
- **Data_ExistsAlready** - Boolean flag indicating that the package was already downloaded before and there was another attempt
- **Data_GetFileStreamFailed** - source package not available in source
- **Data_GetFileStreamFailedToCreateLocalFolder** - local disk issues causing failure in directory creation
- **Data_GetFileStreamFromPackageFailed** - flag indicating if package was downloaded, but the client can't read it
- **Data_GetFileStreamFromPackageSuccess** - successful attempts to read the package
- **Data_GetFileStreamSuccess** - no disk issues nor configuration issues that don't let the file stream to be read
- **Data_GetRelativePathsFailed** - relative path doesn't point to accessible location
- **Data_HashActualValue** - hash value extracted from file name when the package was used
- **Data_HashCalculationFailed** - error with calculation of a hash
- **Data_HashMatchFailed** - hash mismatch between the package name and registry values cached
- **Data_HashMatchSuccess** - hash consistency check success
- **Data_PackageDownloadRequestFailed** - can't download the package
- **Data_PackageDownloadRequestNoData** - downloaded package contains no data
- **Data_PackageDownloadRequestSuccess** - successful download of a package
- **Data_PackageExplodedSuccess** - unpacking attempts statuses

- **Data_PackageOpenFailed** - failed attempts to open the package file
- **Data_PackageOpenSuccess** - successful attempts to open the package file
- **Data_SuccessHashValue** - hash value extracted from file name when the package was downloaded
- **Data_SuccessSource** - surface for which the package was downloaded

Office.Visio.VisioSKU

Captures Visio SKU whether it's standard or professional. Essential to categorize issues based on SKU.

The following fields are collected:

- **Data_VisioSKU:integer** - 0 for Standard SKU and 1 for Professional SKU

Office add-in configuration subtype

Software add-ins and their settings.

Excel.AddinDefinedFunction.CustomFunctionsAllInOne

Collects information about runtime behavior of custom add-in functions. Maintains counters of execution attempts, successful completions, infrastructure errors, and user code errors. This is used to identify reliability issues in the product and fix user-impacting problems.

The following fields are collected:

- **AsyncBegin** - number of async functions that begin
- **AsyncEndAddinError** - number of async functions that end in error
- **AsyncEndInfraFailure** - number of async functions that end with in infra failure
- **AsyncEndSuccess** - number of async functions that end successfully
- **AsyncRemoveCancel** - number of async functions that were canceled
- **AsyncRemoveRecycle** - number of async functions that were removed due to recycle
- **StreamingCycles1** - streaming cycle counter

Office.Extensibility.AppCommands.AppCmdProjectionStatus

Collects information to track which Office add-in installations successfully updated the ribbon vs failed.

Used to fix common registration issues where add-ins are not installed properly and never show up resulting in them being unusable.

The following fields are collected:

- None

Office.Extensibility.AppCommands.AddSolution

Collects install information for Office add-ins that customize the ribbon. Used to detect issues with how custom add-ins modify the Office ribbon.

The following fields are collected:

- **AppVersion** - version of the app
- **SolutionId** - ID of the solution
- **StoreType** - indicates the origin of the app

Office.Extensibility.Catalog.ExchangeGetEntitlements

Data regarding the success for failure of retrieving add-in entitlement data for the Office 365 tenant admin

assigned add-ins. Used for health metrics, charts, and analysis of customer problems.

The following fields are collected:

- **CachingResult** - the result of the attempt to save the service call return value
- **ClientParameter** - client identifier sent in the service call
- **EntitlementsCount** - The number of entitlements expected in the call response
- **EntitlementsParsed** - the number of entitlements parsed from the response
- **IsAllEntitlementsParsed** - whether expected entitlements count matches parsed entitlements count
- **ServiceCallHResult** - the result returned by the service call API
- **TelemetryId** - a GUID representing a unique user
- **UsedCache** - whether the cached response was used instead of making a service call
- **VersionParameter** - Office version number sent in the service call

Office.Extensibility.Catalog.ExchangeGetLastUpdate

Data regarding the success for failure of retrieving the need for updated data regarding the Office 365 tenant admin assigned add-ins. Used for health metrics, charts, and analysis of customer problems. ExchangeGetLastUpdate will always run on boot as part of the host code and determine whether add-in assignments have changed for a user. If so then osf.DLL will be loaded so we can call ExchangeGetEntitlements to get the specific assignments (and ExchangeGetManifests will be called to retrieve any new manifests that are needed). ExchangeGetEntitlements (and ExchangeGetManifests) could also be called on demand after host application has been running. The idea is to not load the large DLL if we don't need to. Without this event in Required, we would not be able to tell if users are failing to get add-ins assigned to them if that first service call fails. It's also the main signal for any auth problems we face talking to our service.

The following fields are collected:

- **Abort** - whether the host was shut down during the service call
- **AllowPrompt** - whether auth prompting was allowed
- **AuthScheme** - the auth scheme requested by exchange
- **BackEndHttpStatus** - http code reported when talking to exchange back end
- **BackupUrl** - the secondary exchange URL to call
- **BEServer** - the back-end exchange server name
- **CalculatedBETarget** - The full name of the exchange back end machine
- **Call(n)_TokenAuthError** - the auth error of the nth service call attempt
- **Call(n)_TokenIsValid** - whether the auth token on the nth service attempt was valid
- **CallMethod** - a string indicating which path the code took
- **ConfigSvcReady** - whether the config service had been initialized yet
- **Date** - value returned by exchange server
- **DiagInfo** - information returned by exchange server
- **End_TicketAuthError** - any error in getting the auth ticket after service call
- **End_TokenIsValid** - whether the auth ticket is valid after the service call

- **EndingIdentityState** - identity objects reported state after making the service calls
- **EwsHandler** - value returned from exchange
- **FEServer** - the exchange front end servicing the request
- **HResult** - the result code
- **HttpStatus** - Http status code returned from exchange
- **IsSupportedAuthResponse** - whether the auth type is one we can use
- **LastUpdateValueRegistry** - hash value retrieved from the registry
- **LastUpdateValueRetrieved** - hash value returned from the service call
- **MSDiagnostics** - value returned from exchange
- **MsoHttpResult** - the enumerator value returned from the http API
- **NeedRefresh** –- This is a true or false field indicating whether the add-in data was stale and we needed to update it.
- **PrimaryUrl** - the main URL to make the service call to
- **RequestId** - value returned from exchange
- **RequestTryCount** - the number of times we attempted to make the service call
- **RequestTryCount** - the number of times we tried to talk to exchange
- **ResultChain** - the series of result code from each of the service call attempts
- **StartingIdentityState** - identity objects reported state before making service calls
- **TelemetryId** - a GUID representing a unique user whether we need to make other service calls

Office.Extensibility.Catalog.ExchangeGetManifests

Data regarding the success for failure of retrieving add-in manifests data for the Office 365 tenant admin assigned add-ins. Used for health metrics, charts, and analysis of customer problems.

The following fields are collected:

- **CachedManifestsParsed** – the number of manifests found in the cache
- **IsAllReturnedManifestsParsed** – whether all the manifests that were returned were able to be parsed
- **ManifestsRequested** – the number of manifests needed
- **ManifestsReturned** – the number of manifest returned from the server
- **ManifestsToRetrieve** – the number of manifests to get from the server
- **ReturnedManifestsParsed** – the number of manifests returned from the server that were successfully parsed
- **TelemetryId** – a GUID representing a unique user

Office.Extensibility.UX.FEnsureLoadOsfDLL

Tracks failure to load Osf.DLL. Detect DLL load failure that prevents feature from running.

The following fields are collected:

- None

Office.Extensibility.UX.FEnsureLoadOsfUIDLL

Tracks failure to load OsfUI.DLL. Detect DLL load failure that prevents feature from running.

The following fields are collected:

- None

Office.Extensibility.UX.FEnsureOsfSharedDLLLoad

Tracks failure to load OsfShared.DLL. Detect DLL load failure that prevents feature from running.

The following fields are collected:

- None

Office.Extensibility.VBATelemetryComObjectInstantiated

Collects information about invocation of automation server or client in VBA solutions. Used to understand interaction between VBA and COM Objects.

The following fields are collected:

- **ComObjectInstantiatedCount** – number of COM Object instantiations
- **HashComObjectInstantiatedClsid** – hash of COM Object Class Identifier
- **HashProjectName** – hash of the VBA project name
- **TagId** – unique tag

Office.Extensibility.VBATelemetryDeclare

Collects information about invocation of Win32 APIs in VBA solutions. Used to understand interaction between VBA and usage and to supplement security investigations.

The following fields are collected:

- **DeclareCount** – number of declarations
- **HashDeclare** – hash of the DLL name
- **HashEntryPoint** – hash of the API Name
- **HashProjectName** – hash of the VBA project name
- **IsPtrSafe** – whether the declaration statement is compatible for different architecture
- **TagId** – unique tag

Office.Outlook.Desktop.Add-ins.Add-inLoaded

Collects the success and failure of Outlook loading of an add-in. This data is actively monitored in order to ensure Outlook is correctly working with customer add-ins. This data is used to detect and investigate issues.

The following fields are collected:

- **Standard HVA activity** with no custom payload

Office.Outlook.Mac.AddinAPIUsage

Collects success and failure of add-in execution in Outlook. This data is actively monitored to ensure Outlook is correctly working with add-ins. This data is used to detect and investigate issues.

The following fields are collected:

- **AccountType** - type of account associated with the add-in
- **Cookie** - cookie used by add-in

- **DispId** - dispatch identifier
- **EndTime** - time when add-in ended
- **ExecutionTime** - time elapsed during execution of add-in
- **Result** - result of using the add-in in Outlook
- **StartTime** - time when add-in started

Office.Outlook.Mac.AddinEventAPIsUsage

Collects success or failure of add-in execution in Outlook. This data is actively monitored to ensure Outlook is correctly working with add-ins. This data is used to detect and investigate issues.

The following fields are collected:

- **AddinType** - type of add-in
- **EventAction** - action performed by the add-in
- **EventDispid** - dispatch identifier
- **EventResult** - result of the action performed by the add-in

Office.Outlook.Mac.AddInInstallationFromInClientStore

Collects success or failure of add-in installation in Outlook. This data is actively monitored to ensure Outlook is correctly working with add-ins. This data is used to detect and investigate issues.

The following fields are collected:

- **AccountType** - type of account associated with add-in
- **FailureReason** - reason add-in failed to install
- **MarketplaceAssetId** - store add-in identifier
- **Status** - status of add-in installation

Office.Programmability.Addins.InternalSetConnectEnterprise

Event generated when a COM Add-in is loaded on an enterprise device. Used to determine adoption, performance, and reliability issues with Office add-ins.

The following fields are collected:

- **Activity Result** - Success state of the connection *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AddinConnectFlag** – represents load behavior
- **AddinDescriptionV2** - the add-in description
- **AddinFileNameV2** - the add-in file name, excluding file path
- **AddinFriendlyNameV2** - the add-in friendly name
- **Add-inId** – the add-in Class ID *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AddinIdV2** - the add-in Class ID
- **AddinProgIdV2** - the add-in prog ID
- **AddinProviderV2** - the add-in provider

- **Add-inTimeDateStamp** – the add-in timestamp from the DLL metadata *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AddinTimeDateStampV2** - the add-in timestamp from the DLL metadata
- **AddinVersionV2** - the add-in version
- **IsBootInProgress** – whether the Office application is in the process of booting
- **LoadDuration** - duration of the add-in load
- **LoadResult** - success state of the load
- **OfficeArchitecture** - Architecture of the Office client

Office.Programmability.Addins.RibbonButtonClick

The event is generated the first time in a session the user clicks a button added to the ribbon by a specific add-in. If the session spans multiple days, this telemetry will be sent once per day instead. The data is used in two ways: 1. When an add-in breaks, knowing how much users truly use the add-in will help us triage the issue. 2. To display to admins as part of COM add-in management scenarios in add-in Inventory and as part of planned add-in health scenarios in Microsoft 365 Apps health. Admins will be able to monitor add-in usage per device, letting them disable or uninstall unused COM add-ins.

The following fields are collected:

- **Add-inTimeDateStamp** - The add-in timestamp from the DLL metadata
- **CLSID** - The add-in class identifier
- **Description** - The add-in description
- **FileName** - The add-in file name, excluding the file path
- **FriendlyName** - The add-in friendly name
- **OfficeApplication** - The Office application currently executing
- **ProgID** - The add-in Prog identifier

Office.Visio.Visio.AddonLoad

Captures errors when a solution fails to load. Essential to debug addon load errors in Visio.

The following fields are collected:

- **Data_Load1Error:integer** - Error value during the load of Visio add-on

Office.Visio.Visio.AddonUsage

Captures errors when there is an error in solution functionality. Essential to debug addon errors in add-ons.

The following fields are collected:

- **Data_DocumentSessionLogID:string** - Document session identifier
- **Data_IsEnabled:bool** - true if solution is enabled
- **Data_TemplateID:string** - GUID of template in which solution was loaded. Logged as 0 for custom solution
- **Data_AddOnID:string** - GUID to identify addon loaded
- **Data_Error:integer** - Error ID

Security subtype

Document, feature, and add-in error conditions that may compromise security, including product update readiness.

Office.AppGuard.CreateContainer

We collect error codes and whether the container already existed or not. We also collect error codes for a reset event in case we fail to create the container on our first attempt. Data will be used identify the percentage of sessions we successfully create the container for launching Office Application Guard apps. Data will also allow Microsoft to identify and address error codes from the container creation.

The following fields are collected:

- **ErrorCode1** - Type of container setup error code.
- **ErrorCode2** - Error code from executing the creation.
- **ErrorCode3** - Additional error code.
- **Id** - A unique identifier (GUID) for the container creation.
- **ResetError** - Error code from trying to reset the container after a failed attempt.
- **ResetErrorCode1** - Type of Container Setup Error code after reset command.
- **ResetErrorCode2** - Error code from executing the creation after reset command.
- **ResetErrorCode3** - Additional error code after reset command.
- **ResetErrorType** - Type of error during reset: Creation, Preparing File or Launch.
- **WarmBoot** - Identifies whether the container was already created or not.

Office.AppGuard.LaunchFile

This event denotes the result of an Application Guard launch file execution. We will be able to define the percentage of sessions we successfully launched a Word, Excel, or PowerPoint file and the error codes for the failed attempts.

The following fields are collected:

- **AppId** – Identifies which App is being launched.
- **DetachedDuration** – Identifies the total time the merged activity took.
- **ErrorCode1** – Type of container setup error code.
- **ErrorCode2** – Error code from executing the creation.
- **ErrorCode3** - Additional error code.
- **FileId** - A unique identifier (GUID) returned from the Windows API after launching a file.
- **Id** – A unique identifier (GUID) for the launching and creating a file. This ID is used to correlate events from Office and Windows.
- **ResetError** - Error code from trying to reset the container after a failed attempt.
- **ResetErrorCode1** – Type of container setup error code after reset command.
- **ResetErrorCode2** – Error code from executing the creation after reset command.
- **ResetErrorCode3** - Additional error code after reset command.
- **ResetErrorType** - Type of error: Creation, PrepFile or Launch.

Office.Security.ActivationFilter.CLSIDActivated

Tracks when a specific Class Identifier (Flash, Silverlight etc.) is activated in Office. Used to track impact of blocking Flash, Silverlight, and Shockwave controls on end users.

The following fields are collected:

- **ActivationType** - Type of activation for the control
- **Blocked** - was the control blocked by office
- **CLSID** - class identifier of the control
- **Count** - how many times the control was activated

Office.Security.ActivationFilter.FailedToRegister

Tracks an error condition in security mitigation that blocks activation of dangerous controls in Office.

Used to diagnose error conditions in security mitigation that could impact security of end users.

The following fields are collected:

- None

Office.Security.ComSecurity.FileExtensionListFromService

Tracks if packager block extensions were read from config service or we used client default list. Used to ensure effectiveness of security mitigation that protects end users of Office.

The following fields are collected:

- **RetrievedFromServiceStatus** - were we able to retrieve the list of file extensions to block if not then what was the error reason

Office.Security.ComSecurity.Load

Tracks when an OLE object is loaded in a document. Used to ensure effectiveness of security mitigation that protects end users of Office.

The following fields are collected:

- **Clsid** - class identifier of the OLE control
- **Count** - how many times the OLE control was loaded
- **DocUrlHash** - a hash representing the document uniquely. (Note – no way to find out the actual details of the document from this. It is just a unique representation of the document.)
- **IsCategorized** – was the OLE control categorized to load in office
- **IsInsertable** – is the OLE control insertable or not

Office.Security.ComSecurity.ObjDetected

Tracks when an OLE object is detected in a document. Used to ensure effectiveness of security mitigation that protects end users of Office.

The following fields are collected:

- **Clsid** - class identifier of the OLE control
- **Count** - how many times this OLE object was detected
- **DocUrlHash** - a hash representing the document uniquely. (Note – no way to find out the actual details of the document from this. It is just a unique representation of the document.)
- **IsCategorized** - is the OLE control categorized to load in office
- **IsInsertable** - is the OLE control insertable or not

Office.Security.ComSecurity.PackagerActivation

Tracks the outcome of security mitigation that blocks dangerous extensions embedded in Office documents. Used to ensure effectiveness of security mitigation that protects end users of Office.

The following fields are collected:

- **FromInternet** - was the document from internet
- **PackagerSetting** - what was the current packager setting
- **TrustedDocument** - was the document a trusted document

Office.Security.ComSecurity.PackagerActivationErrors

Tracks error conditions in security mitigation that blocks dangerous extensions embedded in Office documents. Used to ensure effectiveness of security mitigation that protects end users of Office.

The following fields are collected:

- **Error** - error code
- **Extension** - what was the file extension
- **FromInternet** - was the document from internet
- **LinkedDocument** - was it a linked document or not
- **PackagerSetting** - what was the current packager setting
- **TrustedDocument** - was the document trusted

Office.Security.Macro.InternetVBABlockEnabled

Tracks whether the Block Macro from Internet setting is enabled in a client. Assess use of security mitigation to protect against malicious macros.

The following fields are collected:

- None

Office.Security.Macro.PolicyDigSigTrustedPublishers

Tracks if the macro was verified to be from a trusted publisher. Used to ensure effectiveness of security mitigation that protects end users of Office.

The following fields are collected:

- **Policy** - is the policy set, not set, or not available

Office.Security.Macro.Prompted

Tracks when a user is prompted to enable VBA Macros. Used to assess prevalence of VBA Macros and drive future security mitigations that restrict macro execution in response to security incidents.

The following fields are collected:

- **PromptType** - what type of prompt was shown
- **VBAMacroAntiVirusHash** - antivirus hash of the macro
- **VBAMacroAntiVirusHRESULT** - result of the antivirus assessment

Office.Security.Macro.VBASecureRuntimeSessionEnableState

Tracks each AMSI runtime verification performed when a macro executes. Tracks effectiveness of the AMSI runtime verification of Macro execution and identify errors that could impact security of end users.

The following fields are collected:

- **IsRegistry** - did admin set some override in registry
- **State** - what is the state to for secure runtime

Office.Security.Macro.XL4Prompted

Tracks when a user is prompted to enable XL4 Macros. Used to assess prevalence of XL4 Macros in Excel to drive future security mitigations that block XL4 by default in response to security incidents that involve abusing XL4 macros.

The following fields are collected:

- **PromptType** - what type of prompt was shown

Office.Security.OCX.UFIPrompt

Tracks when a security prompt is shown to the user when loading an ActiveX control that is marked Unsafe for Initialization. Used to track prevalence of UFI ActiveX controls in Office documents to drive mitigations (for example, killbitting controls) in response to security incidents.

The following fields are collected:

- **IsFromInternet** - is the document opened from internet
- **IsSecureReaderMode** - is the document opened in secure reader
- **OcxTrustCenterSettings** - what is the current ActiveX setting

Office.Security.SecureReaderHost.OpenInOSR

Tracks completion of an open in Protected View. Used to diagnose conditions that lead to failures when opening files in Protected View impacting security and productivity of customers.

The following fields are collected:

- None

Product and service usage data events

The following are the data subtypes in this category:

- Application feature success
- Application status and boot
- Office accessibility configuration
- Privacy

Application feature success subtype

Success of application functionality. Limited to opening and closing of the application and documents, file editing, and file sharing (collaboration).

account.action

Needed to ensure account configuration is operating successfully and is used to monitor health of account creation, ability to add new email accounts, and monitor soft account resets

The following fields are collected:

- **account_calendar_count** - how many calendars the account has
- **action** - type of action performed, for example, create_account, delete_account.
- **duration_seconds** - duration of the action
- **entry_point** - entry point of the action, how the user started the action

- **has_hx** - whether the device has an account that is using our new mail syncing service, not necessarily the account that the action was performed upon
- **is_hx** - is the account using our new mail syncing service
- **is_shared_mailbox** - whether the action pertains to a shared mailbox
- **number_of_accounts** - total number of accounts that the action is performed on
- **result** - result for the action, for example, success, failure.
- **server_type** - the server type for the account, similar to account_type
- **shared_type** - type of shared account (if the account is shared)
- **scope** - the scope of the action; for delete account, this_device or all_devices
- **total_calendar_accounts** - count of calendar accounts in the app at time of action
- **total_email_accounts** - count of email accounts in the app at time of action
- **total_file_accounts** - count of file accounts in the app at time of action

account.lifecycle

This event is collected to ensure account configuration is operating successfully and is used to monitor health of account creation, ability to add new email accounts, and monitor soft account resets.

The following fields are collected:

- **account_creation_source** – optional property that is used to find and diagnose any issues that happen during account creation when the action type is add. It can have values like single sign-on (SSO), create_new_account, manual, etc.
- **action** - The type of action performed on the account, such as add, remove, or reset

add.new.account.step

This event lets us detect how far the user has gotten in the create new account form. It indicates when the user has moved to another step or if they have dropped off. We need this information to detect if any steps are failing and to ensure user account creation was successful.

The following field is collected:

- **OTAddAccountCurrentStep** - That can have the following values: profile_form, redirect_mobile_check, mobile_check_success

app.error

Tracks critical app errors used so that we can prevent issues that could cause your app to crash or prevent you from reading email.

The following fields are collected:

- **clientName** - The name of the client for the cloud file where the error occurred, if applicable.
- **cloudfile_error_type** - The type of error that occurred for the cloud file, if applicable.
- **cloudfile_response_name** - The response name for the error that occurred for the cloud file, if applicable.
- **component_name** - The name of the component of the app where the error occurred, such as mail or calendar.
- **debug_info** - Information on the error that occurred for the cloud file in order to be able to determine why the error happened.

- **error_origin_identifier** - Origin of the error that occurred on the draft that the error occurred, if applicable.
- **error_type** - The type of error that occurred. Some examples include save draft, send draft, and cloud file error.
- **exdate** - the extended rule date (only applies to appointment recurrence errors) *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **exrule** - the extended rule value (only applies to appointment recurrence errors) *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **has_attachments** - Reflects if the draft the error occurred on has attachments, if applicable.
- **is_IRM_protected** - Reflects if the draft the error occurred on is protected by information rights management, if applicable.
- **is_legitimate** - Reflects if the error comes from a programming error or not. Programming errors are considered non-legitimate.
- **is_local** - Reflects if the draft the error occurred on has synced to the server, if applicable.
- **is_recoverable** - Reflects if the error can be recovered from or if it is a fatal error.
- **rdate** - the date of the recurrence rule (only applies to appointment recurrence errors) *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **rrule** - the recurrence rule itself (only applies to appointment recurrence errors) *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **rrule_error_message** - recurrence rule parsing error message (only applies to appointment recurrence errors) *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **rrule_error_type** - recurrence rule parsing error type (only applies to appointment recurrence errors) *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **status_code** - The status code of the error that occurred. This helps us understand the cause of the error.

All characters are also possible properties. This helps us understand the characters in the body of the draft message when the error occurred. For example, "a", "b", "c" are possible properties.

app.launch.report

This event is triggered when Outlook starts slowly or incompletely. The data collected provides information on the specific features that were enabled and how long parts of the startup took. It allows us to detect and fix the cause of the issues.

The following fields are collected:

- **is_agenda_widget_active** - Tells us if the agenda widget is active.
- **is_alert_available** - Tell us if the app has been configured to allow alerts in notifications.
- **is_background_refresh_available** - Tells if the app has been configured to be able to refresh in the background.
- **is_badge_available** - Tell us if the app has been configured to allow badges in notifications.
- **is_intune_managed** - Tell us if the app is managed by Intune.
- **is_registered_for_remote_notifications** - Tell us if the app has been registered for remote notifications.

- **is_sound_available** - Tell us if the app has been configured to allow sounds in notifications.
- **is_watch_app_installed** - Tells us if the watch Outlook app has been installed.
- **is_watch_paired** - Tell us if the watch Outlook app is paired with the main Outlook app.
- **launch_to_db_ready_ms** - Tell us the amount of time the Outlook app spent from launch to the database being ready.
- **num_calendar_accounts** - Tells us the number of calendar accounts in the app.
- **num_cloud_file_accounts** - Tells us the number of storage accounts in the app.
- **num_hx_calendar_accounts** - Tells us the number of calendar accounts in the app that connect to our new mail syncing service.
- **num_hx_mail_accounts** - Tells us the number of mail accounts in the app that connect to our new mail syncing service.
- **num_mail_accounts** - Tells us the number of mail accounts in the app.

calendar.action

Used for monitoring any possible negative impact on your ability to perform core calendar actions like creating or editing events. The event could also include a series of property names and if they have changed or not. For example, "title_changed", "online_meeting_changed", and "description_changed" are property names that are included to help us understand if there are any issues with editing certain properties.

The following fields are collected:

- **account_sfb_enabled** - Helps us make sure that Skype for Business is configured correctly.
- **action** - The type of action that was performed on the calendar. This includes things like open, editing, adding shortcut, snooze, etc. Helps us ensure our calendar experience is functioning as expected and nothing has broken
- **action_result** - Result of the action taken on calendar components. This can include values such as success, failure, unknown, and timeout. Used to track the success rate of actions and determine if there is a widespread issue with calendar actions.
- **attendee_busy_status** - The free/busy status of the attendee that the action is related to. This value can be free, busy, or tentative. Helps us understand if there is an issue with actions related to a certain busy status.
- **availability** - The availability value if the free/busy value has changed on the meeting. Helps us understand if there are issues with setting a certain availability value.
- **calendar_onlinemeeting_default_provider** - Contains the default online meeting provider for use with server-supported online meeting creation. This includes types of Skype, Skype for Business, Hangout, and Teams for Business. Helps us diagnose potential issues with creating online meetings on certain providers.
- **calendar_onlinemeeting_enabled** - True if the calendar supports server-supported online meeting creation based on a default online meeting provider. Helps us understand if there are any issues with online meeting enabled calendars.
- **calendar_type** - The type of calendar an event is on after the user has edited the meeting. Possible values include primary, secondary, shared, and group. Helps us understand if there are issues with a certain calendar type.
- **delete_action_origin** - The origin of the delete action performed. This includes values such as

navigation bar toolbar and capsule toolbar. Helps us understand if there are any issues with deleting a meeting from a certain location.

- **distribution_list_count** - Number of attendees that are on distribution lists. Helps us track if there are issues with attendees that are on distribution lists.
- **guest_count** - The number of guests on the meeting. Helps us make sure that the guests are being added correctly.
- **is_all_day** - Used along with "meeting_duration" to specify if this is an all-day meeting. Helps us understand if there are any issues with actions performed on all-day meetings.
- **is_every_meeting_online_on** - True if the users account is set to have online meetings on by default. Helps us understand if there are any issues with online meeting enabled calendars.
- **is_location_permission_granted** – Whether user has granted system location permission to the app. If location permission is granted, the app can show extra utility information in the user interface. Knowing if location permission is granted will allow us to know how often the extra utility information is being shown to users.
- **is_organizer** - Helps us understand if meetings are able to be edited and created by the organizer correctly.
- **is_recurring** - Helps us understand if there is an issue that specifically impacts recurring meetings.
- **launch_point** - The launch point of the action. Can be values such as widget header, widget footer, widget all day, and calendar shortcut. Helps us understand the context that the action was started from.
- **location_count** - Number of locations that are set on event create and edit. Helps us understand if there are any issues with creating or editing events with a certain number of locations.
- **location_selection_source_type** - Type of the location selection source. This can include values such as location suggestion, custom, and existing. Helps us diagnose any issues with selecting a location from a certain source.
- **location_session_id** - ID of the meeting location chooser session. Helps us diagnose any issues with choosing a location to add to the meeting.
- **location_type** - The type of location selected. Contains types such as custom location, conference room, and Bing. Helps us understand issues with adding certain location types to the meeting.
- **meeting_duration** - The length of the meeting. Helps us make sure that the meetings are being configured with the correct times.
- **meeting_insights_type** - The type of meeting insights in the event details. This includes file and message. Helps us understand the number of meeting insights that are being shown.
- **meeting_type** - The type of online meeting associated with the action. This includes types of Skype, Skype for Business, Hangout, and Teams for Business. Helps us understand if the online meetings are configured correctly.
- **online_meeting_provider_switch_type** - The type of switch done by the user between the online meeting providers. Helps us to understand the user's engagement with the feature.
- **origin** - The origin of the calendar action. This includes types like agenda, calendar, widget agenda, etc. Helps us better ensure the interaction within the calendar components are working correctly
- **recurrence_scope** - The type of recurrence of the meeting, either occurrence or series. Helps us understand if there are any issues with editing different meeting recurrence types.

- **reminder_time** - The reminder time of the meeting if it has changed. Used to make sure the reminder time of the meeting is saved correctly.
- **reminders_count** - Number of reminders on the event if the reminders have changed. Helps us diagnose any issues with multiple reminders on an event.
- **sensitivity** - The sensitivity of the meeting. This includes types of normal, personal, private, and confidential. Helps us understand if the sensitivity of the meeting is being configured correctly.
- **session_duration** - The length that the session lasted in milliseconds. Helps us understand if there are issues that are increasing the amount of time needed to perform the calendar action.
- **shared_calendar_result** - The result of an action performed on a shared calendar. Possible values include ok, no permission, unknown, owner on-prem, and owner is group. Helps us understand the reliability of actions performed on shared calendars.
- **time_picker_origin** - Origin of the time picker for a save action. Includes values such as more options and fewer options. Helps us understand how the user navigated the flow to save the meeting and ensure that is functioning correctly
- **title** - The auto-suggested title from app-defined values. This includes values such as "Call", "Lunch", and "Skype". Helps us understand if the title auto-suggestion is configured correctly.
- **txp** - The type of booking or reservation on the event, if any. This includes types like event reservation, flight reservation, car rental reservation, etc. Helps us understand if the booking/reservation cards are performing correctly.
- **upcoming_event_count** - The number of upcoming events displayed in the upcoming events view. Helps us understand if there are issues with the upcoming events view.
- **upcoming_event_seconds_until_event** - The number of seconds until the next upcoming event starts. Helps us understand the typical events that are shown in the upcoming events view.
- **value** - Action-specific detail such as alert delay length or repeat-until category. Helps us understand the context that the action was performed.

combined.search.use

This event is triggered when a user enters or exits search mode or interacts with search entities such as results, suggestions, or filters. Used for monitoring possible negative impact on your ability to perform key search functionality such as searching for mail, contacts, or events.

The following fields are collected across iOS and Android:

- **account_switcher_action_type** - This action type tracks if the user used the account switcher either in simply discovery or if they decided to switch the account
- **action** - the type of action that was performed for search. This identifies if a search has been started, in occurring, or ended and what actions were happening during the search, for example, was the mic used. This is instrumental in ensuring accurate and helpful searches.
- **action_type** - The type of action that was performed for search. This identifies if a search has been started, in occurring, or ended and what actions were happening during the search, for example, was the mic used. This is instrumental in ensuring accurate and helpful searches. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **conversation_id** - Unique ID for every search session (for example, every time the user enters the search box)
- **entrance_type** - This determines how the user started the search query, from the search tab, zero query,

search heading, or search result.

- **has_contact_results** - Simple whether contact results are shown or not in the search query
- **include_deleted** - whether the search is showing deleted options in the search results
- **is_best_match_suggestion** - whether the search suggestion selected is a best match.
- **is_ics_external_data** - Captures if an added event is internal (i.e added in Outlook to Outlook calendar) or external (i.e added from another email app such as Gmail to Outlook calendar).
- **is_network_fully_connected** - This is to get a hint of the reason behind the offline search. If the network is connected and the search is offline, then the reason is likely to be the server timeout
- **is_offline_search** - whether the search session is offline search based on search results returned by hx
- **re_enter_search_tab** - Boolean to indicate whether a user has switched tabs before selecting a search result
- **result_selected_type** - What type of data that was displayed is the user interacting with, for example, see all contact, conversations, event, etc.
- **search_conversation_result_data** - This contains data about the conversation selected from a search result including account type (hx, ac, etc.), whether the message is held by a cloud service, and whether the page offset shown is the same page as the first message.
- **search_origin** - Where did the search originate from, for example, voice assistant, Cortana, keyboard input, etc.
- **search_scope** - A string indicating what type of account the user was searching in (for example, Exchange, Gmail, etc.) or if it was in All Accounts.
- **search_suggestion_type** - indicates what is behind the search suggestion, for example, is a spell correction? Based on history? Autocomplete?
- **search_request_reason** - Indicates the reason a search request was sent from the app, in effect indicating the component or user action that invoked a search.
- **search_result_filter_type** - Indicates what type of filter was applied to search, show all or attachments only

The following fields are collected across iOS applications of Outlook Mobile:

- **all_paging_gesture_count** - Tracks how many search paging gestures were performed in the All tab within the combined search session.
- **all_paging_timeout_count** - Tracks how many paging search requests were blocked due to Hx session timeout in the All tab within the combined search session.
- **answer_result_selected_count** - tracks how many times the search was "successful", for example, did the user find the person they wanted? Composed an email? Bookmarked the message?
- **contact_result_in_full_list_selected_count** - tracks how many times the user asked to "see all contacts" in full list was selected during the combined search session
- **contact_result_selected_count** - tracks how many contact results were selected during the combined search session
- **conversation_result_selected_count** - tracks how many conversations were selecting during the combined search session
- **mail_paging_gesture_count** - tracks how many mail search paging gestures were performed within

the combined search session

- **mail_paging_timeout_count** - Tracks how many paging search requests were blocked due to Hx session timeout in the Mail tab within the combined search session.
- **mail_requests_count** - tracks how many mail search requests were sent within the combined search session
- **people_filter_selected_contacts_count** - tracks how many contacts were selected in the people filter
- **search_session_ended_type** - Indicates where a search ended because it was canceled or the updated was the query
- **search_suggestion_type** - indicates what is behind the search suggestion, for example, is a spell correction? Based on history? Autocomplete?
- **see_all_contacts_selected_count** - tracks how many times "see all contacts" was selected during the combined search session
- **subtab_type** - tracks where the user has selected the result from which result tab
- **top_mail_result_selected_count** - tracks how many times a user selects the top results provided to them.
- **ui_reload_result_count** - records the times of reloading UI because of result set update (during the corresponding query)
- **ui_reload_result_time** - records the total time spent on reloading UI because of result set update (during the corresponding query)
- **ui_reload_status_count** - records the times of reloading UI because of status update (during the corresponding query)
- **ui_reload_status_time** - records the total time spent on reloading UI because of status update (during the corresponding query)

compose.mail.accessory

This event lets us detect and fix issues with key mail compose actions to prevent you from running into issues with attaching a file, taking a photo as an attachment, or sending your availability.

The following fields are collected:

- **action** - Tells us the action that was attempted when the action is logged. Some examples include attaching a file and presenting more options.
- **icon_name** - Tells us the name of the icon that is being shown when the action is logged.
- **origin** – Tells us the origin of the action. Possible values are quick_reply and full_screen.
- **toolbar_type** – Tell us the toolbar type that is presenting on compose page. Possible values are compose_actions and formatting.

conversation.view.action

This event is triggered when a user interacts with the conversation view. For example, attempting to load more conversations or opening quick reply. The data is used for monitoring possible negative impact on the ability to view and reply to email messages as well as for determining issues related to conversation features such as reactions, suggested replies, pinning, etc.

The following fields are collected:

- **contains_mention** - Tells us if the conversation had an @ mention applied to help us detect issues with

email mentions

- **conversation_type** - Tells us what type of email message view was rendered, such as a single message view or multiple message views. Helps us detect issues related to a specific message type in our email conversation view.
- **from_message_reminder** - If the action taken was on a message from a message reminder
- **hx_error_type** - tells us what error occurred that prohibited the service from completing a remove, update, or add reaction on a message.
- **hx_string_tag** - tells us the tag of the error in the service's codebase
- **is_pinned** - Tells us if the conversation is pinned. This is to assess if users are interacting with pin messages and if the pinning feature is behaving as expected.
- **reaction_origin** – Tells us origin from where the user reacted
- **reaction_type** – Tells us the reaction type of the user
- **suggested_file_selected** - Report a Boolean value representing if the user selected a file from the mini-picker
- **suggested_file_shown** - Report a Boolean value representing if file suggestions were shown in the mini-picker
- **suggested_file_time_to_select** - Report the time in ms from when the user clicks the suggested file pill to when they are returned to the compose canvas
- **suggested_reply_char_count** - Tells us how many characters the suggested replies we offer (if available) to help us detect anomalies and issues related to our suggestions
- **suggested_reply_click_pos** - Tells us which position the suggested reply (if available) is rendered so we can detect issues with a specific suggestion
- **suggested_reply_type** - indicates type of suggested reply for this action. Possible values are text, send_avail, and create_meeting.
- **use_default_quick_reply_mode** - Tells us if the default quick reply mode was used to help us detect issues related to the quick reply experience for email

draft.action

Used for monitoring possible negative impact on your ability to create and save mail drafts.

The following fields are collected:

- **action** - the type of action, for example, save, discard.
- **draft_message_id** - message ID of the draft
- **is_groups** - whether the draft is being sent to/from a group folder
- **origin** - where draft was initiated, for example, message detail, compose.
- **smart_compose_model_version** - tracks which version of smart compose model is being used
- **suggestions_requested** - indicates how many smart compose suggestions requested
- **suggestions_results** - smart compose suggestions' result, that is, accepted, rejected
- **suggestions_returned** - indicates how many smart compose suggestions returned from server
- **suggestions_shown** - indicates how many smart compose suggestions shown to the user

- **thread_id** - thread ID of the conversation draft is associated with

drag.and.drop

This event lets us detect if the drag and drop action was successful or not. It is used to ensure that drag-and-drop experiences are working correctly across applications both as a drop event into Outlook and a drag event that leaves Outlook. With this data, we are able to ensure that the end-to-end experience with other applications is working as expected.

The following fields are collected:

- **action** - Action will be either drag or drop
- **location** - In case of a drag action, this will let us know from which location the user started the drag. In case of a drop action, this will let us know where the user dropped the file that was being dragged.
- **source** – In the case of a drop action, this will let us know from which location the user started the drag. This helps us better discover issues with a specific source like OneDrive or Files into a specific drop location, such as a new email.

drawer.event

Used for monitoring possible negative impact on your ability to access folders in your inbox

The following fields are collected:

- **add_calendar_option** - Indicates the type of calendar being added from the drawer that is, interesting calendar, mail calendar, shared calendar, to help us detect issues related to specific calendar types
- **calendar_accounts_count** - Indicates the number of calendar accounts to help us detect issues related to number of accounts you have
- **calendar_apps_count** - Indicates the number of calendar apps present on the user's device to help us detect issues related to calendar apps
- **drawer_type** - Indicates the drawer type: calendar, mail or zero query to help us detect issues related to the drawer type
- **from_favorites** - Indicates if the action was taken from favorites to help us detect issues related to favorites
- **group_calendar_count** - Indicates the number of calendars for the account to help us detect issues related to group calendars
- **inbox_unread_count** - Indicates the number of unread messages in the inbox to help us detect issues with displaying inbox unread counts.
- **interesting_calendar_accounts_count** - Indicates the number of accounts that are eligible for interesting calendars on the device to help us detect issues related to interesting calendars
- **is_group_calendar** - Indicates if the calendar is a group calendar to help us detect issues related to group calendars
- **mail_folder_type** - Indicates the mail folder type that is, inbox, drafts, etc. to help us detect issues related to folder types.
- **mail_accounts_count** - indicates the number of mail accounts to help us detect issues related to mail accounts.
- **selected_group_calendar_count** - Indicates the number of group calendars that are selected and active in the UI
- **visibility_toggle** - indicates if the user is turning on or off a given calendar to help us detect issues

related to showing or hiding calendars

IpcCreateRepublishingLicense

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when the IpcCreateRepublishingLicense API call is made.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.Duration** - Total time for API call to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the API call
- **RMS.HttpCall** - indicate if there is HTTP operation
- **RMS.Result** - Success or fail of the API call
- **RMS.ScenarioId** - Scenario ID defined by the API
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.StatusCode** - Status code of the returned result

IpcGetLicenseProperty

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when the IpcGetLicenseProperty API call is made.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.Duration** - Total time for API call to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the API call
- **RMS.HttpCall** - Indicates if there is HTTP operation
- **RMS.LicensePropertyType** - license property type
- **RMS.Result** - Success or fail of the API call

- **RMS.ScenarioId** - Scenario ID defined by the API
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.StatusCode** - Status code of the returned result

IpcGetSerializedLicenseProperty

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when the IpcGetSerializedLicenseProperty API call is made.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.Duration** - Total time for API call to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the API call
- **RMS.HttpCall** - Indicates if there is HTTP operation
- **RMS.LicensePropertyType** - License property type
- **RMS.Result** - Success or fail of the API call
- **RMS.ScenarioId** - Scenario ID defined by the API
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.StatusCode** - Status code of the returned result

IpcGetTemplateIssuerList

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when the IpcGetTemplateIssuerList API call is made.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.AuthCallbackProvided** - Indicate if provides the authentication callback as input of the API call or not
- **RMS.ConnectionInfo.ExtranetUrl** - extranet URL of connection info
- **RMS.ConnectionInfo.IntranetUrl** - intranet URL of connection info

- **RMS.ConnectionMode** - The connection mode between Rights Management Service client and server: online or offline
- **RMS.Duration** - Total time for API call to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the API call
- **RMS.GuestTenant** - Guest tenant ID for the user
- **RMS.HomeTenant** - Home tenant ID for the user
- **RMS.HttpCall** - indicate if there is HTTP operation
- **RMS.Identity.ExtranetUrl** - The extranet URL of Rights Management service server for the user, collected while getting a new Rights Account Certificate from the server
- **RMS.Identity.IntranetUrl** - The intranet URL of Rights Management service server for the user, collected while getting a new Rights Account Certificate from the server
- **RMS.Identity.Status** - The first time to get Rights Account Certificate from the server or renew the Rights Account Certificate
- **RMS.Identity.Type** - The type of the user account such as windows account or live account
- **RMS.Identity.UserProvided** - Indicate if the user email address provided or not while getting new Rights Account Certificate from the server
- **RMS.IssuerId** - The ID of the Rights Management Service server that issues Rights Account Certificate
- **RMS.LicenseFormat** - The license Format: Xrml or Json
- **RMS.RACType** - The type of Rights Accounts Certificate
- **RMS.Result** - Success or fail of the API call
- **RMS.ScenarioId** - Scenario ID defined by the API
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.ServerType** - The type of Rights Management Service Server
- **RMS.StatusCode** - Status code of the returned result
- **UserInfo.UserObjectId** - The user object ID

IpcGetTemplateList

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when the IpcGetTemplateList API call is made.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID

- **RMS.AuthCallbackProvided** - Indicate if provides the authentication callback as input of the API call or not
- **RMS.ConnectionInfo.ExtranetUrl** - extranet URL of connection info
- **RMS.ConnectionInfo.IntranetUrl** - intranet URL of connection info
- **RMS.ConnectionMode** - The connection mode between Rights Management Service client and server: online or offline
- **RMS.Duration** - Total time for API call to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the API call
- **RMS.GuestTenant** - Guest tenant ID for the user
- **RMS.HomeTenant** - Home tenant ID for the user
- **RMS.HttpCall** - indicate if there is http operation
- **RMS.Identity.ExtranetUrl** - The extranet URL of Rights Management service server for the user, collected while getting a new Rights Account Certificate from the server
- **RMS.Identity.IntranetUrl** - The intranet URL of Rights Management service server for the user, collected while getting a new Rights Account Certificate from the server
- **RMS.Identity.Status** - The first time to get Rights Account Certificate from the server or renew the Rights Account Certificate
- **RMS.Identity.Type** - The type of the user account such as windows account or live account
- **RMS.Identity.UserProvided** - Indicate if the user email address provided or not while getting new Rights Account Certificate from the server
- **RMS.IssuerId** - The ID of the Rights Management Service server that issues Rights Account Certificate
- **RMS.LicenseFormat** - The license Format: Xrml or Json
- **RMS.RACType** - The type of Rights Accounts Certificate
- **RMS.Result** - Success or fail of the API call
- **RMS.ScenarioId** - Scenario ID defined by the API
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.ServerType** - The type of Rights Management Service Server
- **RMS.StatusCode** - Status code of the returned result
- **RMS.TemplatesCount** - The number of the templates
- **UserInfo.UserObjectId** - The user object ID

IpcpCreateLicenseFromScratch

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when the IpcpCreateLicenseFromScratch API call is made.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.Duration** - Total time for API call to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the API call
- **RMS.GuestTenant** - Guest tenant ID for the user
- **RMS.HomeTenant** - Home tenant ID for the user
- **RMS.HttpCall** - indicate if there is HTTP operation
- **RMS.Identity.ExtranetUrl** - The extranet URL of Rights Management service server for the user, collected while getting a new Rights Account Certificate from the server
- **RMS.Identity.IntranetUrl** - The intranet URL of Rights Management service server for the user, collected while getting a new Rights Account Certificate from the server
- **RMS.Identity.UserProvided** - Indicate if the user email address provided or not while getting new Rights Account Certificate from the server
- **RMS.IssuerId** - The ID of the Rights Management Service server that issues Rights Account Certificate
- **RMS.LicenseFormat** - The license Format: Xrml or Json
- **RMS.RACType** - The type of Rights Accounts Certificate
- **RMS.Result** - Success or fail of the API call
- **RMS.ScenarioId** - Scenario ID defined by the API
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.ServerType** - The type of Rights Management Service Server
- **RMS.StatusCode** - Status code of the returned result
- **RMS.TokenProvided** - Indicate if provides the token as input of the API call or not
- **RMS.UserProvided** - Indicate if provides the consumer as input of the API call or not
- **UserInfo.UserObjectId** - The user object ID

IpcpCreateLicenseFromTemplate

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when the IpcpCreateLicenseFromTemplate API call is made.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.

- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.AuthCallbackProvided** - Indicate if provides the authentication callback as input of the API call or not
- **RMS.ConnectionMode** - The connection mode between Rights Management Service client and server: online or offline
- **RMS.Duration** - Total time for API call to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the API call
- **RMS.HttpCall** - indicate if there is http operation
- **RMS.Result** - Success or fail of the API call
- **RMS.ScenarioId** - Scenario ID defined by the API
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.StatusCode** - Status code of the returned result
- **RMS.TokenProvided** - Indicate if provides the token as input of the API call or not
- **RMS.UserProvided** - Indicate if provides the consumer as input of the API call or not

IpcpGetTemplateListForUser

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when the IpcpGetTemplateListForUser API call is made.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.ApplicationScenarioId** - Scenario ID provided by the application
- **RMS.AuthCallbackProvided** - Indicate if provides the authentication callback as input of the API call or not
- **RMS.ConnectionInfo.ExtranetUrl** - extranet URL of connection info
- **RMS.ConnectionInfo.IntranetUrl** - intranet URL of connection info
- **RMS.ConnectionMode** - The connection mode between Rights Management Service client and server: online or offline
- **RMS.Duration** - Total time for API call to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.

- **RMS.ErrorCode** - The error code returned if any from the API call
- **RMS.GuestTenant** - Guest tenant ID for the user
- **RMS.HomeTenant** - Home tenant ID for the user
- **RMS.HttpCall** - Indicates if there is HTTP operation
- **RMS.Identity.ExtranetUrl** - The extranet URL of Rights Management service server for the user, collected while getting a new Rights Account Certificate from the server
- **RMS.Identity.IntranetUrl** - The intranet URL of Rights Management service server for the user, collected while getting a new Rights Account Certificate from the server
- **RMS.Identity.Status** - The first time to get Rights Account Certificate from the server or renew the Rights Account Certificate
- **RMS.Identity.Type** - The type of the user account such as windows account or live account
- **RMS.Identity.UserProvided** - Indicate if the user email address provided or not while getting new Rights Account Certificate from the server
- **RMS.IssuerId** - The ID of the Rights Management Service server that issues Rights Account Certificate
- **RMS.LicenseFormat** - The license Format: Xrml or Json
- **RMS.RACType** - The type of Rights Accounts Certificate
- **RMS.Result** - Success or fail of the API call
- **RMS.ScenarioId** - Scenario ID defined by the API
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.ServerType** - The type of Rights Management Service Server
- **RMS.StatusCode** - Status code of the returned result
- **RMS.TemplatesCount** - The number of the templates
- **RMS.TokenProvided** - Indicate if provides the token as input of the API call or not
- **RMS.UserProvided** - Indicate if provides the consumer as input of the API call or not
- **UserInfo.UserObjectId** - The user object ID

IpcpSerializeLicense

Collected when a user attempts to apply IRM protections on the doc. It contains the information needed to be able to properly investigate and diagnose issues that happen when the IpcpSerializeLicense API call is made.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.ApplicationScenarioId** - Scenario ID provided by the application
- **RMS.AuthCallbackProvided** - Indicate if provides the authentication callback as input of the API call or

not

- **RMS.ConnectionMode** - The connection mode between Rights Management Service client and server: online or offline
- **RMS.ContentId** - Content ID of the document
- **RMS.Duration** - Total time for API call to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the API call
- **RMS.GuestTenant** - Guest tenant ID for the user
- **RMS.HomeTenant** - Home tenant ID for the user
- **RMS.HttpCall** - indicate if there is http operation
- **RMS.Identity.ExtranetUrl** - The extranet URL of Rights Management service server for the user, collected while getting a new Rights Account Certificate from the server
- **RMS.Identity.IntranetUrl** - The intranet URL of Rights Management service server for the user, collected while getting a new Rights Account Certificate from the server
- **RMS.Identity.Status** - The first time to get Rights Account Certificate from the server or renew the Rights Account Certificate
- **RMS.Identity.Type** - The type of the user account such as windows account or live account
- **RMS.Identity.UserProvided** - Indicate if the user email address provided or not while getting new Rights Account Certificate from the server
- **RMS.IssuerId** - The ID of the Rights Management Service server that issues Rights Account Certificate
- **RMS.KeyHandle** - The memory address of key handle
- **RMS.LicenseFormat** - The license Format: Xrml or Json
- **RMS.PL.KeyType** – Values of 'Single' or 'Double.' Indicates whether the PL was protected with Single Key Protection or Double Key Protection.
- **RMS.RACType** - The type of Rights Accounts Certificate
- **RMS.Result** - Success or fail of the API call
- **RMS.ScenarioId** - Scenario ID defined by the API
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.ServerType** - The type of Rights Management Service Server
- **RMS.StatusCode** - Status code of the returned result
- **RMS.TokenProvided** - Indicate if provides the token as input of the API call or not
- **RMS.UserProvided** - Indicate if provides the consumer as input of the API call or not
- **UserInfo.UserObjectId** - The user object ID

IpcSetLicenseProperty

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when the

IpcSetLicenseProperty API call is made.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.Duration** - Total time for API call to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the API call
- **RMS.HttpCall** - indicate if there is http operation
- **RMS.LicensePropertyType** - license property type
- **RMS.Result** - Success or fail of the API call
- **RMS.ScenarioId** - Scenario ID defined by the API
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.StatusCode** - Scenario ID defined by the API

link.clicked.action

The event is used to track users' success in viewing a URL in the Edge web view and completing standard web scenarios in that web view without facing errors

The following fields are collected:

- **account_type** – if the Edge web view was launched from an email or event in Outlook, type of the account where the URL came from
- **action** – action performed by the user inside Outlook from the moment they tap on a URL to when they exit that flow (opened the link in the Edge web view, page failed to load in the web view, performed a search in the web view, exit the Edge web view to open the link in a web browser application, etc.)
- **duration** – duration of the user session
- **launch_type** – if the Edge web view was launched, was it from Outlook, from a widget, or from an OS component
- **origin** – if the user performed an action in the Edge web view, origin of that action
- **referrer** – the location of the URL the user tapped on (email, calendar event, TXP card, etc.)
- **search_scope** – if the user performed a search in the Edge web view, scope of that search (All, Images, Videos, etc.)
- **search_subtype** – if the user performed a search in the Edge web view, was it an initial search or a refined search
- **session_summary_page_loaded_count** – number of pages loaded by the user during their session in the Edge web view

- **session_summary_search_count** - number of Bing searches performed by the user during their session in the Edge web view
- **session_summary_session_id** – identifier for the present user session in the Edge web view
- **txp** – if the Edge web view was launched from a TXP card, event type for that card (dining, flight, etc.)
- **txp_component** - if the Edge web view was launched from a TXP card, UI component type for that card

mail.action

Used for monitoring possible negative impact on your ability to perform critical mail actions (like running mail threaded mode, ensuring mail triage actions work) to ensure our app is functioning properly for mail.

The following fields are collected:

- **account** - the account that performed the action *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **action** - tracks what type of action was taking, that is, archive, delete, mark as read, etc.
- **attachment_content_type** - the content type of the downloaded attachment
- **attachment_content_type_with_count** - tracks the number of attachments in email
- **attachment_download_result** - the result (that is, success/failure) for an attachment download action
- **attachment_download_time** - the time for an attachment download action
- **attachment_extn** - the file extension of the downloaded attachment *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **attachment_id** - the system identifier for the downloaded attachment
- **attachment_size** - the size of downloaded attachment
- **domain** - domain of the document being opened
- **duration** - tracks how long the action took as human-readable English string (for example, 1s, 4h)
- **error** - error message associated with the action
- **event_mode** - what type of event mode it was in, groups or others.
- **Extension** - four characters of file extension of link or attachment associated with this action *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **internet_message_id** - tracking message ID
- **is_group_escalation** - indicates whether the message the action was taken on was sent to the user's mailbox because of an escalation (subscribed to group)
- **is_pinned** - Tells us if the conversation is pinned. This is to assess if users are interacting with pin messages and if the pinning feature is behaving as expected.
- **is_rule** - indicates if the mail action done is resetting a focused/other classification
- **is_threaded_mode** - indicates whether the message was in threaded mode or not, that is, how are the messages grouped
- **is_unread** - indicates whether message is unread that the action was taken on
- **left_swipe_setting** - indicates what action was a set to be the left swipe
- **message_id** - server message ID targeted for action, or comma-separated list if more than one item was

in action.

- **message_type** - indicates what type of message type the action was taken on - group or other
- **number_selected** - the number of items the user selected on the message list and took action on during multiple selection mode.
- **origin** - source of action, that is, cell swipe, zero-query, deep link, email view, email list, etc.
- **origin_view** - source view of action, for example, conversation, message, etc.
- **reported_to_msft** - after sending an email to junk (spam) or trash (phishing) they can choose to report their action to Microsoft.
- **retry** - whether the action was retried
- **right_swipe_setting** - indicates what action was a set to be the right swipe
- **shortcut** - indicates if a shortcut was used and what shortcut was used for scheduling a message that is, later, tomorrow, choose time, etc.
- **size** - size of link or attachment associated with this action
- **source_folder** - tracks source folder type when action is indicating to move from one folder to other, that is, to inbox, trash etc.
- **source_inbox** - indicates which inbox the mail action is taking place (that is, focused, other, etc.) state - state of the action, that is, success or point of failure
- **state** - state of the action, that is, success or point of failure
- **target_folder** - indicates target folder type when moving emails from one folder to other
- **thread_id** - thread ID of the conversation targeted for action, or comma-separated list if more than one item was targeted
- **time_taken_to_fetch_access_token** - time taken to fetch a system access token to use for opening a link
- **time_taken_to_fetch_drive_item** - time taken to fetch a OneDrive resource when clicked
- **time_taken_to_fetch_embed_viewer_resource** - time taken to initialize the embedded viewer when opening links
- **time_taken_to_load_embed_viewer** - time taken to initialize the embedded viewer when opening links
- **time_taken_to_load_link** - time a load link action takes to complete
- **time_taken_to_tap_attachment** - the time between opening the message and clicking on the attachment
- **time_taken_to_tap_link** - time the user took between viewing message and clicking a link
- **txp** - indicates if there is a txp type of item associated with the mail that action was taken on, that is, event reservation, flight reservation, etc.
- **type** - document type being opened via link

mail.compose

Used for monitoring possible negative impact on your ability to compose and reply to emails such as running into issues with reply-all, formatting your email, or sending your emails.

The following fields are collected:

- **draft_message_id** - The draft ID of the conversation being created as a draft to help us detect issues related to draft emails
- **from_context_menu** - Tells us whether the compose is originated from context menu actions.
- **from_message_reminder** - Tells us if the message we are writing is in response to a message reminder
- **message_id** - The message ID of the conversation being replied to or forwarded from to help us detect issues related to a specific message
- **origin** - Tells us where the compose originated from, such as from a reply all, a new compose, or quick reply. Helps us detect issues associated with a specific reply origin type.
- **is_group_escalation** - Whether the message is an escalated group message so we can detect compose issues related to groups.
- **is_link** - Tells us if the new draft that was created was done from a link. Helps us detect issues associated with drafts created from links.
- **is_force_touch** - Tells us if the new draft that was created was done from a force touch action. Helps us detect issues associated with drafts created from this specific action.
- **is_groups** - Whether the event was started from the groups' space so we can detect compose issues related to groups.
- **source_inbox** - Tells us the source inbox, such as whether it was a focused or other inbox
- **thread_id** - The thread ID of the conversation being replied to or forwarded from to help us detect issues related to a specific thread

meeting.call.to.action

Used for monitoring possible negative impact on your ability to perform critical meeting actions like creating, editing, and responding to meetings.

The following fields are collected:

- **event_mode** - Indicates whether this event was from a group or not to help us detect issues with group events
- **meeting_id** - A meeting ID that helps us track issues throughout the lifetime of a meeting to help us detect issues with specific meetings
- **meeting_provider** - Indicates the provider for an online meeting, for example, Teams, Skype for Business to help us detect issues with specific online meeting providers
- **notify_type** - Indicates the response type for other account types to help us detect issues with different account types
- **recurrence** - Indicates how often this meeting occurs that is, occurrence or series to help us detect issues with reoccurring meeting series
- **response** - Indicates response type such as accept or decline on certain account types to help us detect issues with responding to events
- **response_message_length** - Indicates how long the message length was to help us detect issues with meeting responses
- **review_time_proposal_action_type** - Indicates a user response new time proposal to help us detect issues with proposing a new time

- **send_response** - Indicates whether a response was sent to help us detect issues sending meeting invite responses
- **txp** - Indicates what type of meeting it was generated from flight reservations and deliveries to help us detect issues with this type of meeting
- **with_message_enabled** - Indicates whether a user can respond with a message to help us detect issues with responding to meeting invites

message.reminder

This event is triggered when a user interacts with a message reminder. A message reminder is a User Interface (UI) element that prompts the user to interact with a message they might have forgotten about and should follow up on. The data is used to determine the optimal UI for showing message reminders and for monitoring the success and impact of the feature.

The following fields are collected across iOS and Android:

- **origin** - Which view is the message reminder is on
- **action** - The type of action taken on the message reminder. This includes actions such as opening the message, dismissing the reminder, turning off the feature, and when the reminder was rendered.

multi.window.launch

This event captures when the user takes action that involves multi-window launch on foldable devices, e.g., compose mail, event, open calendar window. It uses the action to remember such action, e.g., to keep getting the prompt or always launch in new window. The data collected by this event will be used to assess discoverability, effectiveness as well as general user preferences to drive current and future developments of multi window related functionalities.

The following fields are collected:

- **is_remembered** - whether the preference to launch in a new window from the reported location has been saved by user.
- **multi_window_origin** - the location within the app where the interaction to launch another app screen in a new window happens.

notification.center

This event allows us to track when users enter and exit the notification center in addition to the count of unseen notifications. This helps us make sure that the notification center is consistent with all other clients. We also track when a user taps on a notification so we can tell what type it is.

The following fields are collected:

- **action** - the action taken by the user (closed, opened, notification_tapped)
- **message_reminder_available** - True if there is a message reminder available and will be displayed when the notification center is opened
- **type** - the notification type, as of now it will always be reaction
- **unseen_count** - how many notifications in the current view have not been seen before

Office.Android.DocsUI.FileOperations.OpenDocumentMeasurements

This event is collected for Office applications running under Android platform and records when a file open operation takes place. The event helps in keeping the file open operation secure, up- to- date and performing properly. The goal of collecting this data is to continuously improve the file open performance.

The following fields are collected:

- **Data_AppDocsOperationDuration** - The duration spent in sub- layer during a file open operation.
- **Data_AppDuration** - The duration spent in application processing during a file open operation.
- **Data_AppWarmUpGain** - The gain in application boot duration we get because of pre-booting a part of the application beforehand.
- **Data_BootDuration** - The duration of application boot in process of the file open.
- **Data_BootMarkers** – A string value logging the time duration between some function calls when booting the application, in a format with function ID and duration.
- **Data_ClosePreviouslyOpenedMarkers** – In some file open scenarios, closing of a previously opened document takes place before the opening of the current document. This time duration between some of the operations that take place in this case is captured in a string value that has the format <functionId> <functionValue><functionId><functionValue>...
- **Data_Doc_AccessMode** - An enumeration indicating the access mode of the file, for example, read only, read write.
- **Data_Doc_AsyncOpenKind** - An enumeration indicating the type of asynchronous flow used to open the file.
- **Data_Doc_ChunkingType** - An enumeration indicating the type of chunking algorithm of a file.
- **Data_Doc_EdpState** - An enumeration indicating the enterprise data protection state of a file.
- **Data_Doc_Ext** - File extension of the file.
- **Data_Doc_Fqdn** - The server host name of the file.
- **Data_Doc_FqdnHash** - A Globally Unique Identifier (GUID) that uniquely identifies server host name.
- **Data_Doc_IdentityTelemetryId** - A GUID that uniquely identifies the identity used to open a file.
- **Data_Doc_InitializationScenario** - An enumeration indicating the detailed scenario type of a file open operation.
- **Data_Doc_IOFlags** - An enumeration indicating the IO flags of a file open operation, for example, if the file is cached or not.
- **Data_Doc_IsCloudCollabEnabled** - Whether or not the cloud collaboration is enabled for the file.
- **Data_Doc_IsIncrementalOpen** - Whether or not the file was opened via incremental open.
- **Data_Doc_IsOcsSupported** - Whether or not a file supports Office Collaboration Service.
- **Data_Doc_IsOpeningOfflineCopy** - Whether or not a file is opened from an offline cached copy.
- **Data_Doc_IsPrefetched** - Whether or not the file was prefetched before open operation happened.
- **Data_Doc_IsSyncBacked** - Whether or not a cloud file exists locally and is synchronized with the server.
- **Data_Doc_Location** - An enumeration indicating where the file is located, for example, locally or in cloud.
- **Data_Doc_ReadOnlyReasons** - An enumeration indicating the read only reason of a file.
- **Data_Doc_ResourceIdHash** - A GUID that uniquely identifies server resource ID of the file.
- **Data_Doc_RtcType** - An enumeration indicating type of real-time channel (RTC) used by the file.
- **Data_Doc_ServerDocId** - A GUID that uniquely identifies server document ID.

- **Data_Doc_ServerProtocol** - An enumeration indicating the server protocol of a cloud file.
- **Data_Doc_ServerType** - An enumeration indicating the server type of a cloud file.
- **Data_Doc_ServerVersion** - An enumeration indicating the server version of a cloud file.
- **Data_Doc_SessionId** - An integer that is incremented by 1 for each file open operation in a session.
- **Data_Doc_SharePointServiceContext** - A string used to correlate client- side and server- side logs, typically it is a kind of ID.
- **Data_Doc_SizeInBytes** - File size in bytes.
- **Data_Doc_SpecialChars** - An enumeration indicating which kind of special character the file URL has.
- **Data_Doc_UrlHash** - A GUID that uniquely identifies the file URL.
- **Data_Doc_UsedWrsDataOnOpen** - Whether or not the file was opened incrementally using pre-cached WRS data.
- **Data_Doc_WopiServiceId** - A string indicating which service a Web Application Open Platform Interface Protocol (WOPI) file is from.
- **Data_ErrorId_Code** - An error code indicating a failure in the data collection operation
- **Data_ErrorId_Tag** - A tag in the code to help find the point of failure
- **Data_FileOpenFlowMarkers** – Before the file open process begins, there is some pre-processing involved. This time taken for this pre-processing is captured in a string value that has the format <functionId><functionValue><functionId><functionValue>...
- **Data_FirstPartyProviderApp** - If a file open on Word, Excel, or PowerPoint or Office apps is invoked from another Microsoft app, then the name of that provider app is captured here.
- **Data_InclusiveMeasurements** - A string value logging the time duration spent in some function calls, in a format with function tag and duration which includes the duration of sub- function calls.
- **Data_InitializationReason** - An enumeration indicating how the file is opened, for example, UI element, triggered by another app, etc.
- **Data_Measurements** - A string value logging the time duration spent in some function calls, in a format with function tag and duration which excludes the duration of sub- function calls.
- **Data_OfficeMobileInitReason** - An enumeration indicating the entry point of file open.
- **Data_RenderToInSpaceDuration** – The duration between render end and the silhouette/canvas animation.
- **Data_SilhouetteDuration** - The duration of rendering of the file open.
- **Data_TimeSplitMeasurements** - A string value logging the time duration spent in some function calls, in a format with function tag, start timestamp and duration.

Office.Android.DocsUI.PaywallControl.PreSignInFRE

[This event was previously named Office.DocsUI.PaywallControl.PreSignInFRE.]

This is critical usage telemetry for the upsell in the First Run Experience for unsigned users. This event captures the first-run sign-in metrics. The data will be used to infer insights for the pre sign-in and understand if the user is continuing to the next stage in the user flow.

The following fields are collected:

- **EventDate** - Timestamp of the event occurrence

- **FunnelPoint** - Enumerator to indicate where user is in this experiment funnel. The enumerator will tell if user sees the treatment and drops off or not.
- **SessionID** - Globally Unique Identifier to connect events by session

Office.Android.DocsUI.PaywallControl.SkuChooserToggled

Usage telemetry to view how many times user switches between different SKUs before attempting a purchase. Used to understand usage of the SKU chooser and optimize the in app purchase experience in future versions.

The following fields are collected:

- **EventDate** – Timestamp of the event occurrence
- **SessionID** – GUID to connect events by session

Office.Android.DocsUI.PaywallControl.UserImageClicked

[This event was previously named Office.DocsUI.PaywallControl.UserImageClicked.]

This event measures telemetry to see if users are trying to complete an action by clicking on a user avatar. This data will be used to measure how many users interact with the avatar icon to assess the need for a follow-up experience upon tap.

The following fields are collected:

- **EventDate** -Timestamp of the event occurrence
- **SessionID** - Globally Unique Identifier to connect events by session

Office.Android.EarlyTelemetry.ExpansionFilesAvailability

We are enabling Android Package Kit (APK) expansion files for the Office mobile app. APK Expansion files are supplementary resource files that Android app developers can publish along with their app. To understand the reliability of the expansion files, we log a flag indicating whether expansion files are available or not at every boot.

The following fields are collected:

- **Data_ExpansionFilesAvailable** - A Boolean flag that indicates whether APK Expansion files are available on the device at the time of app boot.

Office.Android.EarlyTelemetry.ExpansionFilesDownloader

We are enabling Android Package Kit (APK) expansion files for the Office mobile app. APK Expansion files are supplementary resource files, that Android app developers can publish along with their app. To understand the reliability of our expansion file download mechanism, we are logging a flag indicating whether we are successfully able to download expansion files.

The following fields are collected:

- **Data_DownloadSuccess** - A Boolean flag that indicates whether APK Expansion files download is successful, whenever we attempt a download during app boot.

Office.Android.EarlyTelemetry.NoteCreated

Critical signal that is used to monitor the ability of Sticky Notes users to create notes in the app. Telemetry is used to ensure critical regression detection for OneNote app and service health. If users can't create a note, this would trigger a high severity incident.

The following fields are collected:

- **IsExportable** - A flag indicating whether this event was a result of a user action or not. Should be set to True as NoteCreated is a user-triggered action.

- **NoteLocalId** - Distinguishable unique identifier assigned to a note at the time of a user creates the note within the app.
- **StickyNotes-SDKVersion** - Version number indicating the version of Sticky Notes the user is using. Allows us to identify which versions of the product are showing an issue so that we can correctly prioritize it.

Office.Android.EarlyTelemetry.NoteViewed

Critical signal that is used to monitor the ability of Sticky Notes users to view notes in the app. Telemetry is used to ensure critical regression detection for OneNote app and service health. If users can't view their notes, this would trigger a high severity incident.

The following fields are collected:

- **HasImages** - A flag indicating whether the note viewed has images stored in it.
- **IsExportable** - A flag indicating whether this event was a result of a user action or not. Should be set to True as NoteViewed is a user-triggered action.
- **NoteLocalId** - Distinguishable unique identifier assigned to a note at the time a user creates the note within the app.
- **StickyNotes-SDKVersion** - Version number indicating the version of Sticky Notes the user is using. Allows us to identify which versions of the product are showing an issue so that we can correctly prioritize it.

Office.Android.Intune.IntuneComplianceRequest

This event is collected for Office applications running on Android, including Office mobile, Word, Excel, PowerPoint, and OneNote. The event indicates an attempt to sign in to an Intune licensed organization account where the organization administrator has configured policy to enforce app conditional access. It is used to understand the number of end users who are attempting to use apps under this policy configuration, and is combined with another event, Office.Android.Intune.IntuneComplianceStatus, to ensure the configured policy is enforced.

No data fields are collected.

Office.Android.Intune.IntuneComplianceStatus

This event is collected for Office applications running on Android, including Office mobile, Word, Excel, PowerPoint, and OneNote. The event indicates an attempt to sign in to an Intune licensed organization account where the organization administrator has configured policy to enforce app conditional access. This event indicates the compliance status of the application to which the user has signed-in and is used to investigate failures. It is combined with another event, Office.Android.Intune.IntuneComplianceRequest, to ensure the configured policy is enforced.

The following fields are collected:

- **Data_ComplianceStatus** - Indicates the compliance status of the application during sign-in with a success or failure error code.
  - -1 – Unknown error
  - 0 – The application is compliant with the organization policies
  - 1 – The application is not compliant with the organization policies
  - 2 – Service-related failures
  - 3 – Network-related failures
  - 4 – Application failed to retrieve authentication token
  - 5 – The response has not been yet received from the service
  - 6 – The company portal application needs to be installed

Office.Android.ODWXPSSO.Telemetry

This event helps in understanding with which other Microsoft app in the device, our app got silent single sign on, from which entry point and so on. Also helps in understanding the failure reason for not getting silent single sign on. We get better insights like from which Microsoft app in the device, we are getting single sign on experience. Act upon failures, where single sign on isn't working as expected.

The following fields are collected:

- **AccountType** - Indicates the account type with which single sign on is happening, like personal Microsoft account or work account.
- **EntryPoint** - Indicates the entry point in the app, from where single sign on attempt was initiated.
- **ErrorCode** - Indicates the error code of the single sign on attempt.
- **ErrorDescription** - Indicates the error message of the single sign on attempt.
- **HResult** - Indicates result status code of the single sign on attempt.
- **ProviderPackageId** - Other Microsoft app in the device from which single sign on is happening.

Office.Android.PhoneNumberSignIns

This event helps in understanding if user signed in or signed up with phone number-based account or email based personal Microsoft account. This event helps in knowing count of users signing in or signing up with phone number based personal Microsoft account.

The following fields are collected:

- **EntryPoint** - Indicates the entry point in the app, from where sign-in attempt was initiated.
- **IsEmailMissing** - Is email missing in the account profile information?
- **IsPhoneNumberMissing** - Is phone number missing in the account profile information?
- **UserDecision** - Indicates the choice made by user like sign-in or sign-up or sign in later.

Office.Android.UserSignInDecision

This event helps in understanding at which stage user is dropping in sign in flow, why sign in is failing, how many users are getting signed in successfully from which entry point in the app and so on. This event helps with sign-in funnel data, which helps in understand at which stage users are getting dropped more and so on.

The following fields are collected:

- **AccountType** - Indicates the account type with which sign-in is attempted like personal account or work account.
- **AfterLicensingState** - Indicates the app licensing state after sign-in completed.
- **AllowedEditsWithoutSignIn** - Indicates how many free edits have lapsed before sign-in was attempted.
- **BeforeLicensingState** - Indicates the app licensing state before sign-in attempted.
- **CompletionState** - Indicates the stage of sign-in completion.
- **EntryPoint** - Indicates the entry point in the app, from where sign-in attempt was initiated.
- **HRDAutoAcceleratedSignUpAttemptCount** - Indicates the count of accelerated sign-ups attempted.
- **HRDAutoAcceleratedSignUpQuitCount** - Indicates the count of accelerated sign-ups canceled.
- **HResult** - Indicates result status code of the sign-in operation.

- **IsPhoneOnlyAuthFeatureEnabled** - Is phone number-based sign-in allowed or not?
- **LicenseActivationHResult** - Indicates the status code of license activation attempt.
- **LicenseActivationMessageCode** - Indicates the message code from licensing service.
- **NoFreeEditsTreatmentValue** - Is free edits allowed or not?
- **SignUpAttemptCount** - Indicates the count of sign-ups attempted.
- **StartMode** - Indicates the mode in which sign-in attempt was started.
- **UserDecision** - Indicates the choice made by user like sign-in or sign-up or sign in later.

Office.AppCompat.AppCompat.AgentScanAndUpload

Only collected when end user has enabled Office Telemetry Dashboard. It collects information on when the Office Telemetry Agent is executed. This is only collected when Office Telemetry Dashboard is enabled and is used to determine the health of Office Telemetry agent.

The following fields are collected:

- **Data.AgentExit** - Timestamp of when the Telemetry agent exits successfully
- **Data.AgentScan** - Timestamp of when the Telemetry agent completed a scan successfully
- **Data.AgentUpload** - Timestamp of when the Telemetry agent completes the upload successfully

Office.AppCompat.AppCompat.AgentUpload

Generated on client startup when end user has enabled Office Telemetry Dashboard. It collects information on when the Office Telemetry Agent has uploaded data to the share folder. The primary use of this event is to monitor the health of the Office Telemetry agent and the secondary use of the event is to estimate usage of the Office Telemetry Dashboard.

The following fields are collected:

- **UploadTime** - the timestamp of the last successful upload performed by the Telemetry Agent.

Office.AppCompat.AppCompat.TelemetryDashboardResiliencyCrashLog

Only collected when Office Telemetry Dashboard has been enabled by end user (most likely an admin). It collects the occurrence of Office Add-ins and documents crashes. This is only collected when user has enabled Office Telemetry Dashboard and is used to determine if there is an increased occurrence of add-in or document crashes.

The following fields are collected:

- **Data.CollectionTime** - Timestamp of when a crash event was logged

Office.AppDocs.AppDocs.DocumentOperation

This event is collected for Office applications running on Android, iOS, Universal, or Windows platforms. The event records when a file operation (create/open/save/export/etc.) takes place and it is used to understand and prioritize user-experiences based on the file operation information.

The following fields are collected:

- **Data_AppIdForReportEndBeforeAppKnown** – App ID when not known before report end called on the operation.
- **Data_CanContinueFromOnBeforeOperationBegins** – CanContinue state, before the begin handler is invoked.
- **Data_DetachedDuration** – The duration of detach process of an event.

- **Data_Doc_AccessMode** – An enumeration indicating the access mode of the file, for example, read only, read write.
- **Data_Doc_AsyncOpenKind** – An enumeration indicating the type of asynchronous flow used to open the file.
- **Data_Doc_ChunkingType** – An enumeration indicating the type of chunking algorithm of a file.
- **Data_Doc_EdpState** – An enumeration indicating the enterprise data protection state of a file.
- **Data_Doc_Ext** – The first four characters of the file extension of the file.
- **Data_Doc_Fqdn** – The server host name of the file.
- **Data_Doc_FqdnHash** – A GUID that uniquely identifies server host name.
- **Data_Doc_IdentityTelemetryId** – A one-way hash of the user identity used to perform the open.
- **Data_Doc_InitializationScenario** – An enumeration indicating the detailed scenario type of a file open operation.
- **Data_Doc_IOFlags** – An enumeration indicating the IO flags of a file open operation, for example, if the file is cached or not.
- **Data_Doc_IsCloudCollabEnabled** – Whether or not the cloud collaboration is enabled for the file.
- **Data_Doc_IsIncrementalOpen** – Whether or not the file was opened via incremental open.
- **Data_Doc_IsOcsSupported** – Whether or not a file supports Office Collaboration Service.
- **Data_Doc_IsOpeningOfflineCopy** – Whether or not a file is opened from an offline cached copy.
- **Data_Doc_IsPrefetched** – Whether or not the file was prefetched before open operation happened.
- **Data_Doc_IsSyncBacked** – Whether or not a cloud file exists locally and is synchronized with the server.
- **Data_Doc_Location** – An enumeration indicating where the file is located, for example, locally or in cloud.
- **Data_Doc_ReadOnlyReasons** – An enumeration indicating the read only reason of a file.
- **Data_Doc_ResourceIdHash** – A GUID that uniquely identifies server resource ID of the file.
- **Data_Doc_RtcType** – An enumeration indicating type of real-time channel (RTC) used by the file.
- **Data_Doc_ServerDocId** – A GUID that uniquely identifies server document ID.
- **Data_Doc_ServerProtocol** – An enumeration indicating the server protocol of a cloud file.
- **Data_Doc_ServerType** – An enumeration indicating the server type of a cloud file.
- **Data_Doc_ServerVersion** – An enumeration indicating the server version of a cloud file.
- **Data_Doc_SessionId** – An integer that is incremented by 1 for each file open operation in a session.
- **Data_Doc_SharePointServiceContext** – A string used to correlate client-side and server-side logs, typically it is a kind of ID.
- **Data_Doc_SizeInBytes** – File size in bytes.
- **Data_Doc_SpecialChars** – An enumeration indicating which kind of special character the file URL has.
- **Data_Doc_UrlHash** – A GUID that uniquely identifies the file URL.
- **Data_Doc_UsedWrsDataOnOpen** – Whether or not the file was opened incrementally using pre-

cached WRS data.

- **Data_Doc_WopiServiceId** – A string indicating which service a WOPI (Web Application Open Platform Interface Protocol) file is from.
- **Data_DocumentInputCurrency** – Type of document input used by the operation.
- **Data_DocumentOperation_AppId** – Enumeration value representing the ID of an app.
- **Data_DocumentOperation_EndEventId** – Tag that represents where the operation ended.
- **Data_DocumentOperation_EndReason** – Enumeration value representing the end reason.
- **Data_DocumentOperation_IsReinitialized** – Is reinitializing a document already open.
- **Data_DocumentOperation_isTargetECBeginEC** – Is the target execution context the same as the context opened from.
- **Data_DocumentOperation_ParamsFlags** – Enumeration flags used to start the operation.
- **Data_DocumentOperation_TelemetryReason** – Enumeration representation of the entry point for the open event. Eg- open from MRU or browse, file activation, etc.
- **Data_FileIOInclusiveMeasurements** – A string value logging the time duration spent in some function calls, in a format with function tag and duration which includes the duration of sub-function calls.
- **Data_FileIOMeasurements** – A string value logging the time duration spent in some function calls, in a format with function tag and duration which excludes the duration of sub-function calls.
- **Data_InitializationReason** – Enumeration representation of the specific reason for the operation. Eg- open from a URL or a local file path, create with file picker, copy to file path, export to URL, etc.
- **Data_IsDisambiguateCsiNetworkConnectivityErrorEnabled**.
- **Data_IsNameMissingInUrl** – Indicates if the name was not parsed from the URL.
- **Data_IsPathMissingForLocalFile** – Indicates if this is a local file without a path.
- **Data_IsUnpackedLinkSupportedForOpen** – Indicates if unpackable link is supported for open.
- **Data_LinksOpenRightScenario** – Enumeration value for the links open right scenario.
- **Data_OpEndEventId** – Tag that represents where the operation ended.
- **Data_OperationType** – Enumeration representation of the generic type of operation. Eg- create, open, copy, save, etc.
- **Data_RelatedPrevOpTelemetryReason** – Is operation related to previous operation.
- **Data_StopwatchDuration** – Total time for the event.
- **Data_UnpackLinkHint** – Enumeration representing potential user action based on unpack link.
- **Data_UnpackLinkPromptResult** – Enumeration representing response of unpack link prompt.

Office.Apple.ActivatePerpetual

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of the perpetual activation flow as well as investigating causes of failures by reviewing the FailedAt values.

The following fields are collected:

- **Data_FailedAt** - We collect a string representing where in the activate perpetual license flow we failed.

Office.Apple.ActivateSubscription

This event is collected for Office applications running under Apple platforms. We collect information related to the migration from the legacy licensing code stack to the vNext licensing code tack. This is used to monitor the health of the subscription activation flow as well as tracking if this is a migration to licensing vNext and if the primary identity was used.

The following fields are collected:

- **Data_ActivatingPrimaryIdentity** - A true/false value denoting if the primary identity was used.
- **Data_NULSubscriptionLicensed** - A true/false value denoting the state of subscription

Office.Apple.CISAuthTicketWithIdentity

This event is collected for Office applications running under Apple platforms. The event is used for capturing auth token generation failures during InAppPurchase on the Mac (the event logs the error code received). This event is used for detecting and helping troubleshoot auth token generation failures

The following fields are collected:

- **Data_EmptyAuthToken** - We collect a string representing where in the activate perpetual license flow we failed.
- **Data_TicketAuthError** - Error code that indicates the cause of failure
- **Data_ValidIdentity** - If the client has a valid identity

Office.Apple.InAppAssociationActivity

This event is collected for Office applications running under Apple platforms. We collect information related to product association after an in-app purchase. We log which subscription SKU we are associating. This is used to monitor the health of the in-app purchase product associations.

The following fields are collected:

- **Data_ProductID** - The subscription SKU we are trying to associate the product to.

Office.Apple.InAppPurchaseActivity

This event is collected for Office applications running under Apple platforms.

We collect information related to product purchases on the AppStore. We track the result of the purchase (Failure, success, payment issue, etc.), the type of the purchase request (restore, purchase) and the SKU/product being purchased (Microsoft 365 Family, etc.). This data is used for monitoring the health of the in-app purchase flows.

The following fields are collected:

- **Data_ Data_PurchaseResult** - The result of the purchase operation
- **Data_ProductID** - The product being purchased
- **Data_PurchaseRequestType** - The type of purchase request

Office.Apple.InTune

This event is collected for Office applications running under Apple platforms. We collect whether the current session is Intune-managed. This is used to pivot/filter on Intune managed sessions and allows us to investigate potential issues related to Office being run as an Intune-managed application.

The following fields are collected:

- **Data_EventID** - We collect a string representing a code that indicates whether the session is intune-managed.

Office.Apple.Licensing.Mac.LicensingState

This event is collected for Office applications running under Apple platforms. The event captures the current state of the license for a session in a machine (OLS license ID, SKU being used, grace-period or not, RFM, etc.). The data collected is used for detecting errors and investigating causes of failures.

The following fields are collected:

- **Data_DidRunPreview** - A string indicating if this session is run under preview
- **Data_LicensingACID** - A string representing a licensing system internal identifier
- **Data_LicensingType** - A string representing the type of license
- **Data_OLSLicenseId** - A string representing a license identifier
- **Data_State** - A string representing the current state of the license

Office.ConnectDevice.Activity.Start

Allows us to know if a connection to a device or application was successful. Used for feature health and monitoring. This event is generated by Microsoft Data Streamer for Excel Add-in.

The following fields are collected:

- **Datasource_Type** - Serial device, or App Service information
- **DataSource_Name** - Name of connected data source
- **Activity_Name** = Name of the activity "ConnectDevice"
- **Activity_CV** = ID to correlate the events across the connection session
- **Activity_StartStopType** = Start
- **Activity_DateTimeTicks** = Data Time for the activity

Office.ConnectDevice.Activity.Stop

Allows us to know if a connection to a device or application was successful. Used for feature health and monitoring This event is generated by Microsoft Data Streamer for Excel Add-in.

The following fields are collected:

- **Datasource_Type** - Serial device, or App Service information
- **DataSource_Name** - Name of connected data source
- **Activity_Name** - Name of the activity "ConnectDevice"
- **Activity_CV** - ID to correlate the events across the connection session
- **Activity_StartStopType** - Stop
- **Activity_DateTimeTicks** - Data Time for the activity

Office.Docs.AppDocs.OperationOpenFromMruByPath

This event is collected for Office applications running on Android, iOS, Universal, or Windows platforms. The event records when a file open operation takes place from the path provided in the most recently used list and is used to understand and prioritize user-experience errors based on file open operation information.

The following fields are collected:

- **Data_AppIdForReportEndBeforeAppKnown** – App ID when not known before report end called on the operation.
- **Data_CanContinueFromOnBeforeOperationBegins** – CanContinue state, before the begin handler is

invoked.

- **Data_DetachedDuration** – The duration of detach process of an event.
- **Data_Doc_AccessMode** – An enumeration indicating the access mode of the file, for example, read only, read write.
- **Data_Doc_AsyncOpenKind** – An enumeration indicating the type of asynchronous flow used to open the file.
- **Data_Doc_ChunkingType** – An enumeration indicating the type of chunking algorithm of a file.
- **Data_Doc_EdpState** – An enumeration indicating the enterprise data protection state of a file.
- **Data_Doc_Ext** – First 4 characters of the extension of the file.
- **Data_Doc_Fqdn** – The server host name of the file.
- **Data_Doc_FqdnHash** – A GUID that uniquely identifies server host name.
- **Data_Doc_IdentityTelemetryId** – A one-way hash of the user identity used to perform the open.
- **Data_Doc_InitializationScenario** – An enumeration indicating the detailed scenario type of a file open operation.
- **Data_Doc_IOFlags** – An enumeration indicating the IO flags of a file open operation, for example, if the file is cached or not.
- **Data_Doc_IsCloudCollabEnabled** – Whether or not the cloud collaboration is enabled for the file.
- **Data_Doc_IsIncrementalOpen** – Whether or not the file was opened via incremental open.
- **Data_Doc_IsOcsSupported** – Whether or not a file supports Office Collaboration Service.
- **Data_Doc_IsOpeningOfflineCopy** – Whether or not a file is opened from an offline cached copy.
- **Data_Doc_IsPrefetched** – Whether or not the file was prefetched before open operation happened.
- **Data_Doc_IsSyncBacked** – Whether or not a cloud file exists locally and is synchronized with the server.
- **Data_Doc_Location** – An enumeration indicating where the file is located, for example, locally or in cloud.
- **Data_Doc_ReadOnlyReasons** – An enumeration indicating the read only reason of a file.
- **Data_Doc_ResourceIdHash** – A GUID that uniquely identifies server resource ID of the file.
- **Data_Doc_RtcType** – An enumeration indicating type of real-time channel (RTC) used by the file.
- **Data_Doc_ServerDocId** – A GUID that uniquely identifies server document ID.
- **Data_Doc_ServerProtocol** – An enumeration indicating the server protocol of a cloud file.
- **Data_Doc_ServerType** – An enumeration indicating the server type of a cloud file.
- **Data_Doc_ServerVersion** – An enumeration indicating the server version of a cloud file.
- **Data_Doc_SessionId** – An integer that is incremented by 1 for each file open operation in a session.
- **Data_Doc_SharePointServiceContext** – A string used to correlate client-side and server-side logs, typically it is a kind of ID.
- **Data_Doc_SizeInBytes** – File size in bytes.
- **Data_Doc_SpecialChars** – An enumeration indicating which kind of special character the file URL has.

- **Data_Doc_UrlHash** – A GUID that uniquely identifies the file URL.
- **Data_Doc_UsedWrsDataOnOpen** – Whether or not the file was opened incrementally using pre-cached WRS data.
- **Data_Doc_WopiServiceId** – A string indicating which service a WOPI (Web Application Open Platform Interface Protocol) file is from.
- **Data_DocumentInputCurrency** – Type of document input used by the operation.
- **Data_DocumentOperation_AppId** – Enumeration value representing the ID of an app.
- **Data_DocumentOperation_EndEventId** – Tag that represents where the operation ended.
- **Data_DocumentOperation_EndReason** – Enumeration value representing the end reason.
- **Data_DocumentOperation_IsReinitialized** – Is reinitializing a document already open.
- **Data_DocumentOperation_ParamsFlags** – Enumeration flags used to start the operation.
- **Data_DocumentOperation_TelemetryReason** – Enumeration representation of the entry point for the open event. Eg- open from MRU or browse, file activation, etc.
- **Data_DocumentOperation_isTargetECBeginEC** – Is the target execution context the same as the context opened from.
- **Data_FileIOInclusiveMeasurements** – A string value logging the time duration spent in some function calls, in a format with function tag and duration which includes the duration of sub-function calls.
- **Data_FileIOMeasurements** – A string value logging the time duration spent in some function calls, in a format with function tag and duration which excludes the duration of sub-function calls.
- **Data_IsNameMissingInUrl** – Indicates if the name was not parsed from the URL.
- **Data_IsPathMissingForLocalFile** – Indicates if this is a local file without a path.
- **Data_IsUnpackedLinkSupportedForOpen** – Indicates if unpackable link is supported for open.
- **Data_LinksOpenRightScenario** – Enumeration value for the links open right scenario.
- **Data_OpEndEventId** – Tag that represents where the operation ended.
- **Data_RelatedPrevOpTelemetryReason** – Is operation related to previous operation.
- **Data_StopwatchDuration** – Total time for the event.
- **Data_UnpackLinkHint** – Enumeration representing potential user action based on unpack link.
- **Data_UnpackLinkPromptResult** – Enumeration representing response of unpack link prompt.

Office.Docs.AppDocs.OperationOpenFromMruByUrl

This event is collected for Office applications running on Android, iOS, Universal, or Windows platforms. The event records when a file open operation takes place from the URL provided in the most recently used list and is used to understand and prioritize user-experiences based on file open operation information.

The following fields are collected:

- **Data_AppIdForReportEndBeforeAppKnown** – App ID when not known before report end called on the operation.
- **Data_CanContinueFromOnBeforeOperationBegins** – CanContinue state, before the begin handler is invoked.

- **Data_DetachedDuration** – The duration of detach process of an event.
- **Data_Doc_AccessMode** – An enumeration indicating the access mode of the file, for example, read only, read write.
- **Data_Doc_AsyncOpenKind** – An enumeration indicating the type of asynchronous flow used to open the file.
- **Data_Doc_ChunkingType** – An enumeration indicating the type of chunking algorithm of a file.
- **Data_Doc_EdpState** – An enumeration indicating the enterprise data protection state of a file.
- **Data_Doc_Ext** – First 4 characters of the extension of the file.
- **Data_Doc_Fqdn** – The server host name of the file.
- **Data_Doc_FqdnHash** – A GUID that uniquely identifies server host name.
- **Data_Doc_IdentityTelemetryId** – A one-way hash of the user identity used to perform the open.
- **Data_Doc_InitializationScenario** – An enumeration indicating the detailed scenario type of a file open operation.
- **Data_Doc_IOFlags** – An enumeration indicating the IO flags of a file open operation, for example, if the file is cached or not.
- **Data_Doc_IsCloudCollabEnabled** – Whether or not the cloud collaboration is enabled for the file.
- **Data_Doc_IsIncrementalOpen** – Whether or not the file was opened via incremental open.
- **Data_Doc_IsOcsSupported** – Whether or not a file supports Office Collaboration Service.
- **Data_Doc_IsOpeningOfflineCopy** – Whether or not a file is opened from an offline cached copy.
- **Data_Doc_IsPrefetched** – Whether or not the file was prefetched before open operation happened.
- **Data_Doc_IsSyncBacked** – Whether or not a cloud file exists locally and is synchronized with the server.
- **Data_Doc_Location** – An enumeration indicating where the file is located, for example, locally or in cloud.
- **Data_Doc_ReadOnlyReasons** – An enumeration indicating the read only reason of a file.
- **Data_Doc_ResourceIdHash** – A GUID that uniquely identifies server resource ID of the file.
- **Data_Doc_RtcType** – An enumeration indicating type of real-time channel (RTC) used by the file.
- **Data_Doc_ServerDocId** – A GUID that uniquely identifies server document ID.
- **Data_Doc_ServerProtocol** – An enumeration indicating the server protocol of a cloud file.
- **Data_Doc_ServerType** – An enumeration indicating the server type of a cloud file.
- **Data_Doc_ServerVersion** – An enumeration indicating the server version of a cloud file.
- **Data_Doc_SessionId** – An integer that is incremented by 1 for each file open operation in a session.
- **Data_Doc_SharePointServiceContext** – A string used to correlate client-side and server-side logs, typically it is a kind of ID.
- **Data_Doc_SizeInBytes** – File size in bytes.
- **Data_Doc_SpecialChars** – An enumeration indicating which kind of special character the file URL has.
- **Data_Doc_UrlHash** – A GUID that uniquely identifies the file URL.

- **Data_Doc_UsedWrsDataOnOpen** – Whether or not the file was opened incrementally using pre-cached WRS data.
- **Data_Doc_WopiServiceId** – A string indicating which service a WOPI (Web Application Open Platform Interface Protocol) file is from.
- **Data_DocumentInputCurrency** – Type of document input used by the operation.
- **Data_DocumentOperation_AppId** – Enumeration value representing the ID of an app.
- **Data_DocumentOperation_EndEventId** – Tag that represents where the operation ended.
- **Data_DocumentOperation_EndReason** – Enumeration value representing the end reason.
- **Data_DocumentOperation_IsReinitialized** – Is reinitializing a document already open.
- **Data_DocumentOperation_ParamsFlags** – Enumeration flags used to start the operation.
- **Data_DocumentOperation_TelemetryReason** – Enumeration representation of the entry point for the open event. Eg- open from MRU or browse, file activation, etc.
- **Data_DocumentOperation_isTargetECBeginEC** – Is the target execution context the same as the context opened from.
- **Data_FileIOInclusiveMeasurements** – A string value logging the time duration spent in some function calls, in a format with function tag and duration which includes the duration of sub-function calls.
- **Data_FileIOMeasurements** – A string value logging the time duration spent in some function calls, in a format with function tag and duration which excludes the duration of sub-function calls.
- **Data_IsNameMissingInUrl** – Indicates if the name was not parsed from the URL.
- **Data_IsPathMissingForLocalFile** – Indicates if this is a local file without a path.
- **Data_IsUnpackedLinkSupportedForOpen** – Indicates if unpackable link is supported for open.
- **Data_LinksOpenRightScenario** – Enumeration value for the links open right scenario.
- **Data_OpEndEventId** – Tag that represents where the operation ended.
- **Data_RelatedPrevOpTelemetryReason** – Is operation related to previous operation.
- **Data_StopwatchDuration** – Total time for the event.
- **Data_UnpackLinkHint** – Enumeration representing potential user action based on unpack link.
- **Data_UnpackLinkPromptResult** – Enumeration representing response of unpack link prompt.

Office.Docs.AppDocs.OperationOpenFromPath

This event is collected for Office applications running on Android, iOS, Universal, or Windows platforms. The event records when a file open operation takes place from a path and is used to understand and prioritize user-experiences based on file open operation information.

The following fields are collected:

- **Data_AppIdForReportEndBeforeAppKnown** – App ID when not known before report end called on the operation.
- **Data_CanContinueFromOnBeforeOperationBegins** – CanContinue state, before the begin handler is invoked.
- **Data_DetachedDuration** – The duration of detach process of an event.

- **Data_Doc_AccessMode** – An enumeration indicating the access mode of the file, for example, read only, read write.
- **Data_Doc_AsyncOpenKind** – An enumeration indicating the type of asynchronous flow used to open the file.
- **Data_Doc_ChunkingType** – An enumeration indicating the type of chunking algorithm of a file.
- **Data_Doc_EdpState** – An enumeration indicating the enterprise data protection state of a file.
- **Data_Doc_Ext** – First 4 characters of the extension of the file.
- **Data_Doc_Fqdn** – The server host name of the file.
- **Data_Doc_FqdnHash** – A GUID that uniquely identifies server host name.
- **Data_Doc_IdentityTelemetryId** – A one-way hash of the user identity used to perform the open.
- **Data_Doc_InitializationScenario** – An enumeration indicating the detailed scenario type of a file open operation.
- **Data_Doc_IOFlags** – An enumeration indicating the IO flags of a file open operation, for example, if the file is cached or not.
- **Data_Doc_IsCloudCollabEnabled** – Whether or not the cloud collaboration is enabled for the file.
- **Data_Doc_IsIncrementalOpen** – Whether or not the file was opened via incremental open.
- **Data_Doc_IsOcsSupported** – Whether or not a file supports Office Collaboration Service.
- **Data_Doc_IsOpeningOfflineCopy** – Whether or not a file is opened from an offline cached copy.
- **Data_Doc_IsPrefetched** – Whether or not the file was prefetched before open operation happened.
- **Data_Doc_IsSyncBacked** – Whether or not a cloud file exists locally and is synchronized with the server.
- **Data_Doc_Location** – An enumeration indicating where the file is located, for example, locally or in cloud.
- **Data_Doc_ReadOnlyReasons** – An enumeration indicating the read only reason of a file.
- **Data_Doc_ResourceIdHash** – A GUID that uniquely identifies server resource ID of the file.
- **Data_Doc_RtcType** – An enumeration indicating type of real-time channel (RTC) used by the file.
- **Data_Doc_ServerDocId** – A GUID that uniquely identifies server document ID.
- **Data_Doc_ServerProtocol** – An enumeration indicating the server protocol of a cloud file.
- **Data_Doc_ServerType** – An enumeration indicating the server type of a cloud file.
- **Data_Doc_ServerVersion** – An enumeration indicating the server version of a cloud file.
- **Data_Doc_SessionId** – An integer that is incremented by 1 for each file open operation in a session.
- **Data_Doc_SharePointServiceContext** – A string used to correlate client-side and server-side logs, typically it is a kind of ID.
- **Data_Doc_SizeInBytes** – File size in bytes.
- **Data_Doc_SpecialChars** – An enumeration indicating which kind of special character the file URL has.
- **Data_Doc_UrlHash** – A GUID that uniquely identifies the file URL.
- **Data_Doc_UsedWrsDataOnOpen** – Whether or not the file was opened incrementally using pre-

cached WRS data.

- **Data_Doc_WopiServiceId** – A string indicating which service a WOPI (Web Application Open Platform Interface Protocol) file is from.
- **Data_DocumentInputCurrency** – Type of document input used by the operation.
- **Data_DocumentOperation_AppId** – Enumeration value representing the ID of an app.
- **Data_DocumentOperation_EndEventId** – Tag that represents where the operation ended.
- **Data_DocumentOperation_EndReason** – Enumeration value representing the end reason.
- **Data_DocumentOperation_IsReinitialized** – Is reinitializing a document already open.
- **Data_DocumentOperation_ParamsFlags** – Enumeration flags used to start the operation.
- **Data_DocumentOperation_TelemetryReason** – Enumeration representation of the entry point for the open event. Eg- open from MRU or browse, file activation, etc.
- **Data_DocumentOperation_isTargetECBeginEC** – Is the target execution context the same as the context opened from.
- **Data_FileIOInclusiveMeasurements** – A string value logging the time duration spent in some function calls, in a format with function tag and duration which includes the duration of sub-function calls.
- **Data_FileIOMeasurements** – A string value logging the time duration spent in some function calls, in a format with function tag and duration which excludes the duration of sub-function calls.
- **Data_IsNameMissingInUrl** – Indicates if the name was not parsed from the URL.
- **Data_IsPathMissingForLocalFile** – Indicates if this is a local file without a path.
- **Data_IsUnpackedLinkSupportedForOpen** – Indicates if unpackable link is supported for open.
- **Data_LinksOpenRightScenario** – Enumeration value for the links open right scenario.
- **Data_OpEndEventId** – Tag that represents where the operation ended.
- **Data_RelatedPrevOpTelemetryReason** – Is operation related to previous operation.
- **Data_StopwatchDuration** – Total time for the event.
- **Data_UnpackLinkHint** – Enumeration representing potential user action based on unpack link.
- **Data_UnpackLinkPromptResult** – Enumeration representing response of unpack link prompt.

Office.Docs.AppDocs.OperationOpenFromProtocolHandler

This event is collected for Office applications running on Android, iOS, Universal, or Windows platforms. The event records when a file open operation takes place from another application using the protocol handler interface and is used to understand and prioritize user-experiences based on file open operation information.

The following fields are collected:

- **Data_AppIdForReportEndBeforeAppKnown** – App ID when not known before report end called on the operation.
- **Data_CanContinueFromOnBeforeOperationBegins** – CanContinue state, before the begin handler is invoked.
- **Data_DetachedDuration** – The duration of detach process of an event.
- **Data_Doc_AccessMode** – An enumeration indicating the access mode of the file, for example, read only,

read write.

- **Data_Doc_AsyncOpenKind** – An enumeration indicating the type of asynchronous flow used to open the file.
- **Data_Doc_ChunkingType** – An enumeration indicating the type of chunking algorithm of a file.
- **Data_Doc_EdpState** – An enumeration indicating the enterprise data protection state of a file.
- **Data_Doc_Ext** – First 4 characters of the extension of the file.
- **Data_Doc_Fqdn** – The server host name of the file.
- **Data_Doc_FqdnHash** – A GUID that uniquely identifies server host name.
- **Data_Doc_IdentityTelemetryId** – A one-way hash of the user identity used to perform the open.
- **Data_Doc_InitializationScenario** – An enumeration indicating the detailed scenario type of a file open operation.
- **Data_Doc_IOFlags** – An enumeration indicating the IO flags of a file open operation, for example, if the file is cached or not.
- **Data_Doc_IsCloudCollabEnabled** – Whether or not the cloud collaboration is enabled for the file.
- **Data_Doc_IsIncrementalOpen** – Whether or not the file was opened via incremental open.
- **Data_Doc_IsOcsSupported** – Whether or not a file supports Office Collaboration Service.
- **Data_Doc_IsOpeningOfflineCopy** – Whether or not a file is opened from an offline cached copy.
- **Data_Doc_IsPrefetched** – Whether or not the file was prefetched before open operation happened.
- **Data_Doc_IsSyncBacked** – Whether or not a cloud file exists locally and is synchronized with the server.
- **Data_Doc_Location** – An enumeration indicating where the file is located, for example, locally or in cloud.
- **Data_Doc_ReadOnlyReasons** – An enumeration indicating the read only reason of a file.
- **Data_Doc_ResourceIdHash** – A GUID that uniquely identifies server resource ID of the file.
- **Data_Doc_RtcType** – An enumeration indicating type of real-time channel (RTC) used by the file.
- **Data_Doc_ServerDocId** – A GUID that uniquely identifies server document ID.
- **Data_Doc_ServerProtocol** – An enumeration indicating the server protocol of a cloud file.
- **Data_Doc_ServerType** – An enumeration indicating the server type of a cloud file.
- **Data_Doc_ServerVersion** – An enumeration indicating the server version of a cloud file.
- **Data_Doc_SessionId** – An integer that is incremented by 1 for each file open operation in a session.
- **Data_Doc_SharePointServiceContext** – A string used to correlate client-side and server-side logs, typically it is a kind of ID.
- **Data_Doc_SizeInBytes** – File size in bytes.
- **Data_Doc_SpecialChars** – An enumeration indicating which kind of special character the file URL has.
- **Data_Doc_UrlHash** – A GUID that uniquely identifies the file URL.
- **Data_Doc_UsedWrsDataOnOpen** – Whether or not the file was opened incrementally using pre-cached WRS data.

- **Data_Doc_WopiServiceId** – A string indicating which service a WOPI (Web Application Open Platform Interface Protocol) file is from.
- **Data_DocumentInputCurrency** – Type of document input used by the operation.
- **Data_DocumentOperation_AppId** – Enumeration value representing the ID of an app.
- **Data_DocumentOperation_EndEventId** – Tag that represents where the operation ended.
- **Data_DocumentOperation_EndReason** – Enumeration value representing the end reason.
- **Data_DocumentOperation_IsReinitialized** – Is reinitializing a document already open.
- **Data_DocumentOperation_ParamsFlags** – Enumeration flags used to start the operation.
- **Data_DocumentOperation_TelemetryReason** – Enumeration representation of the entry point for the open event. Eg- open from MRU or browse, file activation, etc.
- **Data_DocumentOperation_isTargetECBeginEC** – Is the target execution context the same as the context opened from.
- **Data_FileIOInclusiveMeasurements** – A string value logging the time duration spent in some function calls, in a format with function tag and duration which includes the duration of sub-function calls.
- **Data_FileIOMeasurements** – A string value logging the time duration spent in some function calls, in a format with function tag and duration which excludes the duration of sub-function calls.
- **Data_IsNameMissingInUrl** – Indicates if the name was not parsed from the URL.
- **Data_IsPathMissingForLocalFile** – Indicates if this is a local file without a path.
- **Data_IsUnpackedLinkSupportedForOpen** – Indicates if unpackable link is supported for open.
- **Data_LinksOpenRightScenario** – Enumeration value for the links open right scenario.
- **Data_OpEndEventId** – Tag that represents where the operation ended.
- **Data_RelatedPrevOpTelemetryReason** – Is operation related to previous operation.
- **Data_StopwatchDuration** – Total time for the event.
- **Data_UnpackLinkHint** – Enumeration representing potential user action based on unpack link.
- **Data_UnpackLinkPromptResult** – Enumeration representing response of unpack link prompt.

Office.Docs.AppDocs.OperationOpenFromShell

This event is collected for Office applications running on Android, iOS, Universal, or Windows platforms. The event records when a file open operation takes place from the shell and is used to understand and prioritize user-experiences based on file open operation information.

The following fields are collected:

- **Data_AppIdForReportEndBeforeAppKnown** – App ID when not known before report end called on the operation.
- **Data_CanContinueFromOnBeforeOperationBegins** – CanContinue state, before the begin handler is invoked.
- **Data_DetachedDuration** – The duration of detach process of an event.
- **Data_Doc_AccessMode** – An enumeration indicating the access mode of the file, for example, read only, read write.

- **Data_Doc_AsyncOpenKind** – An enumeration indicating the type of asynchronous flow used to open the file.
- **Data_Doc_ChunkingType** – An enumeration indicating the type of chunking algorithm of a file.
- **Data_Doc_EdpState** – An enumeration indicating the enterprise data protection state of a file.
- **Data_Doc_Ext** – First 4 characters of the extension of the file.
- **Data_Doc_Fqdn** – The server host name of the file.
- **Data_Doc_FqdnHash** – A GUID that uniquely identifies server host name.
- **Data_Doc_IdentityTelemetryId** – A one-way hash of the user identity used to perform the open.
- **Data_Doc_InitializationScenario** – An enumeration indicating the detailed scenario type of a file open operation.
- **Data_Doc_IOFlags** – An enumeration indicating the IO flags of a file open operation, for example, if the file is cached or not.
- **Data_Doc_IsCloudCollabEnabled** – Whether or not the cloud collaboration is enabled for the file.
- **Data_Doc_IsIncrementalOpen** – Whether or not the file was opened via incremental open.
- **Data_Doc_IsOcsSupported** – Whether or not a file supports Office Collaboration Service.
- **Data_Doc_IsOpeningOfflineCopy** – Whether or not a file is opened from an offline cached copy.
- **Data_Doc_IsPrefetched** – Whether or not the file was prefetched before open operation happened.
- **Data_Doc_IsSyncBacked** – Whether or not a cloud file exists locally and is synchronized with the server.
- **Data_Doc_Location** – An enumeration indicating where the file is located, for example, locally or in cloud.
- **Data_Doc_ReadOnlyReasons** – An enumeration indicating the read only reason of a file.
- **Data_Doc_ResourceIdHash** – A GUID that uniquely identifies server resource ID of the file.
- **Data_Doc_RtcType** – An enumeration indicating type of real-time channel (RTC) used by the file.
- **Data_Doc_ServerDocId** – A GUID that uniquely identifies server document ID.
- **Data_Doc_ServerProtocol** – An enumeration indicating the server protocol of a cloud file.
- **Data_Doc_ServerType** – An enumeration indicating the server type of a cloud file.
- **Data_Doc_ServerVersion** – An enumeration indicating the server version of a cloud file.
- **Data_Doc_SessionId** – An integer that is incremented by 1 for each file open operation in a session.
- **Data_Doc_SharePointServiceContext** – A string used to correlate client-side and server-side logs, typically it is a kind of ID.
- **Data_Doc_SizeInBytes** – File size in bytes.
- **Data_Doc_SpecialChars** – An enumeration indicating which kind of special character the file URL has.
- **Data_Doc_UrlHash** – A GUID that uniquely identifies the file URL.
- **Data_Doc_UsedWrsDataOnOpen** – Whether or not the file was opened incrementally using pre-cached WRS data.
- **Data_Doc_WopiServiceId** – A string indicating which service a WOPI (Web Application Open Platform

Interface Protocol) file is from.

- **Data_DocumentInputCurrency** – Type of document input used by the operation.
- **Data_DocumentOperation_AppId** – Enumeration value representing the ID of an app.
- **Data_DocumentOperation_EndEventId** – Tag that represents where the operation ended.
- **Data_DocumentOperation_EndReason** – Enumeration value representing the end reason.
- **Data_DocumentOperation_IsReinitialized** – Is reinitializing a document already open.
- **Data_DocumentOperation_ParamsFlags** – Enumeration flags used to start the operation.
- **Data_DocumentOperation_TelemetryReason** – Enumeration representation of the entry point for the open event. Eg- open from MRU or browse, file activation, etc.
- **Data_DocumentOperation_isTargetECBeginEC** – Is the target execution context the same as the context opened from.
- **Data_FileIOInclusiveMeasurements** – A string value logging the time duration spent in some function calls, in a format with function tag and duration which includes the duration of sub-function calls.
- **Data_FileIOMeasurements** – A string value logging the time duration spent in some function calls, in a format with function tag and duration which excludes the duration of sub-function calls.
- **Data_IsNameMissingInUrl** – Indicates if the name was not parsed from the URL.
- **Data_IsPathMissingForLocalFile** – Indicates if this is a local file without a path.
- **Data_IsUnpackedLinkSupportedForOpen** – Indicates if unpackable link is supported for open.
- **Data_LinksOpenRightScenario** – Enumeration value for the links open right scenario.
- **Data_OpEndEventId** – Tag that represents where the operation ended.
- **Data_RelatedPrevOpTelemetryReason** – Is operation related to previous operation.
- **Data_StopwatchDuration** – Total time for the event.
- **Data_UnpackLinkHint** – Enumeration representing potential user action based on unpack link.
- **Data_UnpackLinkPromptResult** – Enumeration representing response of unpack link prompt.

Office.Docs.AppDocs.OperationOpenFromUrl

This event is collected for Office applications running on Android, iOS, Universal, or Windows platforms. The event records when a file open operation takes place from a URL and is used to understand and prioritize user-experiences based on file open operation information.

The following fields are collected:

- **Data_AppIdForReportEndBeforeAppKnown** – App ID when not known before report end called on the operation.
- **Data_CanContinueFromOnBeforeOperationBegins** – CanContinue state, before the begin handler is invoked.
- **Data_DetachedDuration** – The duration of detach process of an event.
- **Data_Doc_AccessMode** – An enumeration indicating the access mode of the file, for example, read only, read write.
- **Data_Doc_AsyncOpenKind** – An enumeration indicating the type of asynchronous flow used to open

the file.

- **Data_Doc_ChunkingType** – An enumeration indicating the type of chunking algorithm of a file.
- **Data_Doc_EdpState** – An enumeration indicating the enterprise data protection state of a file.
- **Data_Doc_Ext** – First 4 characters of the extension of the file.
- **Data_Doc_Fqdn** – The server host name of the file.
- **Data_Doc_FqdnHash** – A GUID that uniquely identifies server host name.
- **Data_Doc_IdentityTelemetryId** – A one-way hash of the user identity used to perform the open.
- **Data_Doc_InitializationScenario** – An enumeration indicating the detailed scenario type of a file open operation.
- **Data_Doc_IOFlags** – An enumeration indicating the IO flags of a file open operation, for example, if the file is cached or not.
- **Data_Doc_IsCloudCollabEnabled** – Whether or not the cloud collaboration is enabled for the file.
- **Data_Doc_IsIncrementalOpen** – Whether or not the file was opened via incremental open.
- **Data_Doc_IsOcsSupported** – Whether or not a file supports Office Collaboration Service.
- **Data_Doc_IsOpeningOfflineCopy** – Whether or not a file is opened from an offline cached copy.
- **Data_Doc_IsPrefetched** – Whether or not the file was prefetched before open operation happened.
- **Data_Doc_IsSyncBacked** – Whether or not a cloud file exists locally and is synchronized with the server.
- **Data_Doc_Location** – An enumeration indicating where the file is located, for example, locally or in cloud.
- **Data_Doc_ReadOnlyReasons** – An enumeration indicating the read only reason of a file.
- **Data_Doc_ResourceIdHash** – A GUID that uniquely identifies server resource ID of the file.
- **Data_Doc_RtcType** – An enumeration indicating type of real-time channel (RTC) used by the file.
- **Data_Doc_ServerDocId** – A GUID that uniquely identifies server document ID.
- **Data_Doc_ServerProtocol** – An enumeration indicating the server protocol of a cloud file.
- **Data_Doc_ServerType** – An enumeration indicating the server type of a cloud file.
- **Data_Doc_ServerVersion** – An enumeration indicating the server version of a cloud file.
- **Data_Doc_SessionId** – An integer that is incremented by 1 for each file open operation in a session.
- **Data_Doc_SharePointServiceContext** – A string used to correlate client-side and server-side logs, typically it is a kind of ID.
- **Data_Doc_SizeInBytes** – File size in bytes.
- **Data_Doc_SpecialChars** – An enumeration indicating which kind of special character the file URL has.
- **Data_Doc_UrlHash** – A GUID that uniquely identifies the file URL.
- **Data_Doc_UsedWrsDataOnOpen** – Whether or not the file was opened incrementally using pre-cached WRS data.
- **Data_Doc_WopiServiceId** – A string indicating which service a WOPI (Web Application Open Platform Interface Protocol) file is from.

- **Data_DocumentInputCurrency** – Type of document input used by the operation.
- **Data_DocumentOperation_AppId** – Enumeration value representing the ID of an app.
- **Data_DocumentOperation_EndEventId** – Tag that represents where the operation ended.
- **Data_DocumentOperation_EndReason** – Enumeration value representing the end reason.
- **Data_DocumentOperation_IsReinitialized** – Is reinitializing a document already open.
- **Data_DocumentOperation_ParamsFlags** – Enumeration flags used to start the operation.
- **Data_DocumentOperation_TelemetryReason** – Enumeration representation of the entry point for the open event. Eg- open from MRU or browse, file activation, etc.
- **Data_DocumentOperation_isTargetECBeginEC** – Is the target execution context the same as the context opened from.
- **Data_FileIOInclusiveMeasurements** – A string value logging the time duration spent in some function calls, in a format with function tag and duration which includes the duration of sub-function calls.
- **Data_FileIOMeasurements** – A string value logging the time duration spent in some function calls, in a format with function tag and duration which excludes the duration of sub-function calls.
- **Data_IsNameMissingInUrl** – Indicates if the name was not parsed from the URL.
- **Data_IsPathMissingForLocalFile** – Indicates if this is a local file without a path.
- **Data_IsUnpackedLinkSupportedForOpen** – Indicates if unpackable link is supported for open.
- **Data_LinksOpenRightScenario** – Enumeration value for the links open right scenario.
- **Data_OpEndEventId** – Tag that represents where the operation ended.
- **Data_RelatedPrevOpTelemetryReason** – Is operation related to previous operation.
- **Data_StopwatchDuration** – Total time for the event.
- **Data_UnpackLinkHint** – Enumeration representing potential user action based on unpack link.
- **Data_UnpackLinkPromptResult** – Enumeration representing response of unpack link prompt.

Office.Docs.Apple.DocsUXiOSSaveAsThroughFileMenu

This event is collected for Office applications running under Apple platforms. The event records when a "Save as" operation takes place and is used to understand and prioritize user-experiences based on file operation information such as location categories. A "Save as" operation occurs whenever a user creates a new file and saves it for the first time or saves a copy of an existing file to a new location.

The following fields are collected:

- **Data_OriginServiceType** - An abstract categorization of the original location of a file like "SharePoint", "OneDrive", "Local", "WOPI", etc., and explicitly not the actual location of the file.
- **Data_ServiceType** - An abstract categorization of the new location of a file after the save is completed like "SharePoint", "OneDrive", "Local", "WOPI", etc., and explicitly not the actual location of the file.

Office.Docs.Apple.DocsUXMacAtMentionInsertedAtMention

This event is collected for Office applications running under Apple platforms. This event records when a user "@" mentions another user and is used to understand and prioritize user-experiences based on how users collaborate with other users.

The following fields are collected:

- **Data_CharactersTyped** - A numerical value that indicates the total number of characters typed in the "@" mention text.

Office.Docs.Apple.DocsUXMacODSPSharingWebViewSharingCompleted

This event is collected for Office applications running under Apple platforms. This event records when a user chooses to share a cloud document using the OneDrive sharing experience and is used to better understand and prioritize user-experiences based on sharing documents.

The following fields are collected:

- **Data_ShareType** - A hardcoded string that indicates what kind of share operation was completed including but not limited to "Copy Link", "More apps", "Teams".
- **Data_ShareWebViewMode** - A hardcoded string that indicates what kind of share mode was active when the share was completed including but not limited to "ManageAccess", "AtMentions", "Share".

Office.DocsUI.Collaboration.CoauthorGalleryRowTapped

This event is collected for Office applications running under Apple platforms. This event records when a user selects to look at the list of current co-authors. This data is used to better understand and prioritize user-experiences relating to co-authoring a document at the same time.

The following fields are collected:

- **Data_CoauthorCount** - A numerical value that represents the total number of people who are currently editing the same document as the user.

Office.DocsUI.Collaboration.CollabCornerPeopleGalleryCoauthorsUpdated

This event is collected for Office applications running under Apple platforms. The event records when the number of active co-authors in a cloud document changes. This data is used to better understand and prioritize user-experiences relating to co-authoring a document at the same time.

The following fields are collected:

- **Data_CoauthorsJoined** - The number of co-authors that joined the document.
- **Data_CoauthorsLeft** - The number of co-authors that left the document.
- **Data_NewCoauthorCount** - The new count of active co-authors in the document.
- **Data_OldCoauthorCount** - The previous count of active co-authors before the update.
- **Data_ServiceType** - An abstract categorization of the location of a file like "SharePoint", "OneDrive", "Local", "WOPI", etc., and explicitly not the actual location of the file.

Office.DocsUI.DocStage.DocStageCreateNewFromTemplate

This event is collected for Office applications running under Apple platforms. The event records when a new file is created from the "New from template" experience and is used to better understand and prioritize user-experiences based on document creation information.

The following fields are collected:

- **Data_InHomeTab** - A Boolean value that indicates whether the new file from template was created from the Home tab of the file new experience.
- **Data_InSearch** - A Boolean that indicates whether the file was created when the user was searching for a template.
- **Data_IsHomeTabEnabled** - A Boolean value that indicates if the Home tab is currently available to the user.
- **Data_IsRecommendedEnabled** - A Boolean value that indicates if the "Recommended" experience is

currently available to the user.

- **Data_TemplateIndex** - The numerical index of the template file as it is displayed visually to the user.
- **Data_TemplateType** - A classification to help distinguish the type of template like, but not limited to, "Online" templates, "Online search" templates, "Local" templates.

Office.DocsUI.DocStage.RecommendedOpen

This event is collected for Office applications running under Apple platforms. The event records when a file-open operation takes place from the recommended files section of the document gallery and is used to understand and prioritize user-experiences based on file open operation information.

The following fields are collected:

- **Data_Success** - A Boolean value to indicate whether the operation succeeded.

Office.DocsUI.FileOperations.DocsUIFileOpenMacRequired

This event is collected for Office applications running under Apple platforms. The event records when a file open operation takes place and is used to understand and prioritize user-experiences based on file open operation information such as location categories "ServiceType" and the first four characters of the extension.

The following fields are collected:

- **Data_Ext** - The file extension limited to the first four characters of the extension or less.
- **Data_ServiceType** - An abstract categorization of the location of a file like "SharePoint", "OneDrive", "Local", "WOPI", etc.

Office.DocsUI.FileOperations.OpenDocumentMeasurements

This event is collected for Office applications running under iOS platform. The event records when a file open operation takes place and is used to understand and prioritize user-experiences based on file open operation information, especially performance information.

The following fields are collected:

- **Data_AppDuration** - The duration spent in application processing during a file open operation.
- **Data_BootDuration** - The duration of boot process of the file open.
- **Data_ClickOrigin** - A string indicating which part the link was from when user clicked a link in iOS Outlook to open a file in Office app.
- **Data_ClickTime** - The Unix epoch time when the user clicked a link in iOS Outlook to open the file in Office app.
- **Data_ClosePreviouslyOpenedMarkers** – A string value logging the time duration between some function calls, in a format with function ID and duration.
- **Data_DetachedDuration** - The duration of detach process of an event.
- **Data_Doc_AccessMode** - An enumeration indicating the access mode of the file, for example, read only, read write.
- **Data_Doc_AsyncOpenKind** - An enumeration indicating the type of asynchronous flow used to open the file.
- **Data_Doc_ChunkingType** - An enumeration indicating the type of chunking algorithm of a file.
- **Data_Doc_EdpState** - An enumeration indicating the enterprise data protection state of a file.
- **Data_Doc_Ext** - File extension of the file.

- **Data_Doc_Fqdn** - The server host name of the file.
- **Data_Doc_FqdnHash** - A GUID that uniquely identifies server host name.
- **Data_Doc_IdentityTelemetryId** - A GUID that uniquely identifies the identity used to open a file.
- **Data_Doc_InitializationScenario** - An enumeration indicating the detailed scenario type of a file open operation.
- **Data_Doc_IOFlags** - An enumeration indicating the IO flags of a file open operation, for example, if the file is cached or not.
- **Data_Doc_IsCloudCollabEnabled** - Whether or not the cloud collaboration is enabled for the file.
- **Data_Doc_IsIncrementalOpen** - Whether or not the file was opened via incremental open.
- **Data_Doc_IsOcsSupported** - Whether or not a file supports Office Collaboration Service.
- **Data_Doc_IsOpeningOfflineCopy** - Whether or not a file is opened from an offline cached copy.
- **Data_Doc_IsPrefetched** - Whether or not the file was prefetched before open operation happened.
- **Data_Doc_IsSyncBacked** - Whether or not a cloud file exists locally and is synchronized with the server.
- **Data_Doc_Location** - An enumeration indicating where the file is located, for example, locally or in cloud.
- **Data_Doc_ReadOnlyReasons** - An enumeration indicating the read only reason of a file.
- **Data_Doc_ResourceIdHash** - A GUID that uniquely identifies server resource ID of the file.
- **Data_Doc_RtcType** - An enumeration indicating type of real-time channel (RTC) used by the file.
- **Data_Doc_ServerDocId** - A GUID that uniquely identifies server document ID.
- **Data_Doc_ServerProtocol** - An enumeration indicating the server protocol of a cloud file.
- **Data_Doc_ServerType** - An enumeration indicating the server type of a cloud file.
- **Data_Doc_ServerVersion** - An enumeration indicating the server version of a cloud file.
- **Data_Doc_SessionId** - An integer that is incremented by 1 for each file open operation in a session.
- **Data_Doc_SharePointServiceContext** - A string used to correlate client-side and server-side logs, typically it is a kind of ID.
- **Data_Doc_SizeInBytes** - File size in bytes.
- **Data_Doc_SpecialChars** - An enumeration indicating which kind of special character the file URL has.
- **Data_Doc_UrlHash** - A GUID that uniquely identifies the file URL.
- **Data_Doc_UsedWrsDataOnOpen** - Whether or not the file was opened incrementally using pre-cached WRS data.
- **Data_Doc_WopiServiceId** - A string indicating which service a WOPI (Web Application Open Platform Interface Protocol) file is from.
- **Data_HWModel** – A string value logging the model of iPad or iPhone device.
- **Data_InclusiveMeasurements** - A string value logging the time duration spent in some function calls, in a format with function tag and duration which includes the duration of sub-function calls.
- **Data_InitializationReason** - An enumeration indicating how the file is opened, for example, from which

UI element or triggered by another app.

- **Data_IsDocumentAlreadyOpen** – Whether or not the file is already open.
- **Data_IsInterrupted** – Whether or not the file open operation was interrupted by app transitioning to background.
- **Data_Measurements** - A string value logging the time duration spent in some function calls, in a format with function tag and duration which excludes the duration of sub-function calls.
- **Data_OpenInPlace** - Whether or not a file must be copied to the Office's sandboxed container before user can open it.
- **Data_OpenStartTime** - The Unix epoch time when the file open started.
- **Data_PrefetchSourceOptions** - An enumeration indicating how the file is made available offline for cloud documents, e.g., from recent and recommended files.
- **Data_SilhouetteDuration** - The duration of rendering of the file open.
- **Data_SourceApplication** - A string indicating the bundle ID of the source application when a file open was triggered by another app.
- **Data_StopwatchDuration** - The duration from beginning of the event to the end of the event.
- **Data_TimeSplitMeasurements** - A string value logging the time duration spent in some function calls, in a format with function tag, start timestamp and duration.

Office.DocsUI.FileOperations.OpenFileWithReason

This event is collected for Office applications running under Apple platforms. The event records when a file open operation takes place and is used to understand and prioritize user-experiences based on file open operation information such as location categories "ServiceType" and from where within Application the user requested to open a file.

The following fields are collected:

- **Data_IsCandidateDropboxFile** - This is a Boolean value that is logged if by inspecting the path of the file we think it might be from a folder that is synched by Drop Box.
- **Data_IsSignedIn** - Whether or not a user is signed in when the file is saved.
- **Data_OpenReason** - The open reason is a numerical value that indicates from where within the application a user opened a file.
- **Data_ServiceType** - An abstract numerical categorization of the location of a file like "SharePoint", "OneDrive", "Local", "WOPI", etc., and explicitly not the actual location of the file.

Office.DocsUI.FileOperations.SaveToURL

This event is collected for Office applications running under Apple platforms. The event records when a "save as" operation takes place and is used to understand and prioritize user-experiences based on file operation information such as location categories and the first four characters of the extension. A "save as" operation occurs whenever a user creates a new file and saves it for the first time or saves a copy of an existing file to a new location.

The following fields are collected:

- **Data_FileExtension** - The first four characters of the new file's extension.
- **Data_IsNewFileCreation** - Indicates if the save operation is for a new file or a copy of an existing file.
- **Data_IsSignedIn** - Whether or not a user is signed in when the file is saved.

- **Data_SaveErrorCode** - A numerical value that is set if there is an error to help identify the kind of error.
- **Data_SaveErrorDomain** - Specifies the domain of the SaveErrorCode as defined by Apple SaveErrorDomains "are arbitrary strings used to differentiate groups of codes".
- **Data_SaveLocation** - An abstract categorization of the location of a file like "SharePoint", "OneDrive", "Local", "WOPI", etc., and explicitly not the actual location of the file.
- **Data_SaveOperationType** - A numerical value defined by Apple's NSSaveOperationType group of values.

Office.DocsUI.SharingUI.CloudUpsellShown

This event is collected for Office applications running under Apple platforms. This event records when a user goes through the document upsell to cloud flow. This data is used to better understand and prioritize user-experiences relating to moving documents to cloud locations.

The following fields are collected:

- **Data_FileStyle** - A numerical value that indicates from what scenario the upsell experience was shown like from an autosave toggle or a share button.
- **Data_FileType** - The first four characters of the current file's extension.
- **Data_InDocStage** - A Boolean that indicates if the upsell experience is shown from the Document Gallery or from within a document window.
- **Data_IsDocumentOpened** - A Boolean that indicates if the current document for which the upsell experience is being shown is also open.
- **Data_IsDraft** - A Boolean that indicates if the current file has ever been saved.
- **Data_IsSheetModal** - A Boolean that indicates if the upsell experience was presented modally or not.

Office.DocsUI.SharingUI.CloudUpsellUpload

This event is collected for Office applications running under Apple platforms. This event records when a user chooses to upload a new or local file to the cloud and the result of that operation. This data is used to better understand and prioritize user-experiences relating to moving documents to cloud locations.

The following fields are collected:

- **Data_FileStyle** - A numerical value that indicates from what scenario the upsell experience was shown like an autosave toggle or a share button.
- **Data_FileType** - The first four characters of the current file's extension.
- **Data_InDocStage** - A Boolean that indicates if the upsell experience is shown from the Document Gallery or from within a document window.
- **Data_IsDefaultServiceLocation** - A Boolean value that indicates if the selected location to upload the document to is the default location.
- **Data_IsDocumentOpened** - A Boolean that indicates if the current document for which the upsell experience is being shown is also open.
- **Data_IsDraft** - A Boolean that indicates if the current file has ever been saved.
- **Data_IsSheetModal** - A Boolean that indicates if the upsell experience was presented modally or not.
- **Data_LocationServiceType** - An abstract categorization of the location of a file like "SharePoint", "OneDrive", "Local", "WOPI", etc., and explicitly not the actual location of the file.
- **Data_UploadAction** - A hard-coded string that indicates whether the upload was a move or a copy

operation.

- **Data_UploadResult** - A hard-coded string that indicates the result of the attempt to upload including but not limited to ''Success", "UserCancelledUpload", and "PreAuthFailed".

Office.DocsUI.SharingUI.CopyLinkOperation

This event is collected for Office applications running under Apple platforms. This event records when a user chooses to share a document by generating a link to a cloud document and is used to better understand and prioritize user-experiences based on sharing documents.

The following fields are collected:

- **Data_ ServiceType** - An abstract categorization of the location of a file like "SharePoint", "OneDrive", "Local", "WOPI", etc., and explicitly not the actual location of the file.
- **Data_LinkType** - A hard-coded string that describes the kind of invite operation performed like "ViewOnly" and "ViewAndEdit".
- **Data_ShareScenario** - A hard-coded string description of where within the application's user interface the file is being shared from including but not limited to, "FileMenu", "OpenTabShareActionMenu", "RecentTabShareActionMenu".

Office.DocsUI.SharingUI.DocsUIOneDriveShare

This event is collected for Office applications running under Apple platforms. This event records when a user chooses to share a cloud document using the OneDrive sharing experience and is used to better understand and prioritize user-experiences based on sharing documents.

The following fields are collected:

- **Data_ODSPShareWebviewShareError** - If the sharing experience experiences an error this is a numerical value to help identify the reason for the failure.
- **Data_ODSPShareWebviewShareGrantAccessResult** - A Boolean value that when true indicates that a lightweight sharing operation successfully completed.
- **Data_ODSPShareWebviewShareSuccessType** - When a share operation successfully completes this is a numerical value used to determine what kind of sharing operation was completed.
- **Data_WebViewInfoResult** - If the user interface fails to load this is a numerical value to help identify the reason for the failure.
- **Data_WebViewLoadTimeInMs** - A numerical value that records the amount of time it took for the web user interface to load.

Office.DocsUI.SharingUI.InvitePeople

This event is collected for Office applications running under Apple platforms. This event records when a user chooses to invite people to a cloud document and is used to better understand and prioritize user-experiences based on sharing documents.

The following fields are collected:

- **Data_ ServiceType** - An abstract categorization of the location of a file like "SharePoint", "OneDrive", "Local", "WOPI", etc., and explicitly not the actual location of the file.
- **Data_InviteeCount** - The total number of contacts invited to a document in one invite action.
- **Data_LinkType** - A hard-coded string that describes the kind of invite operation performed like "ViewOnly" and "ViewAndEdit".
- **Data_MessageLength** - A numerical count of the total number of characters sent in the invite message.

- **Data_ShareScenario** - A hard-coded string description of where within the application's user interface the file is being shared from including but not limited to, "FileMenu", "OpenTabShareActionMenu", "RecentTabShareActionMenu".

Office.DocsUI.SharingUI.SendACopyOperation

This event is collected for Office applications running under Apple platforms. The event records when a user chooses to send a copy of a document and is used to better understand and prioritize user-experiences based on sharing documents.

The following fields are collected:

- **Data_IsHomeTabEnabled** - A Boolean value that indicates if the Home tab is currently available to the user.
- **Data_IsRecommendedEnabled** - A Boolean value that indicates if the "Recommended" experience is currently available to the user.
- **Data_OperationType** - A numerical value to indicate what kind of send a copy operation is taking place like sending a copy in an email or sending a copy through Apple's share control.
- **Data_ServiceType** - An abstract categorization of the location of a file like "SharePoint", "OneDrive", "Local", "WOPI", etc., and explicitly not the actual location of the file.
- **Data_ShareFileType** - A hard-coded string description of what type of object is being shared including but not limited to, "Document", "PDF", "Picture".
- **Data_ShareScenario** - A hard-coded string description of where within the application's user interface the file is being shared from including but not limited to, "FileMenu", "OpenTabShareActionMenu", "RecentTabShareActionMenu".
- **Data_SharingService** - A Boolean that indicates whether the file was created when the user was searching for a template.

Office.DocsUI.SharingUI.UpsellShare

This event is collected for Office applications running under Apple platforms. This event records when a user goes through the document upsell to cloud flow when trying to share a document. This data is used to better understand and prioritize user experiences relating to moving documents to cloud locations.

The following fields are collected:

- **Data_FileOperationResult** - A numerical value to indicate whether the operation succeeded.
- **Data_HostedFromDocStage** - A Boolean to indicate if a user is going through the upsell to cloud flow from the DocStage experience or from an open document.
- **Data_isLocalCopyOn** - A Boolean to indicate if the use chose to keep a local copy of the document being uploaded to a cloud location or move the existing document to a cloud location.
- **Data_NewFileType** - An abstract categorization of the location of the new location of the file like "SharePoint", "OneDrive", "Local", "WOPI", etc., and explicitly not the actual location of the file.
- **Data_OriginalFileType** - An abstract categorization of the location of a file like "SharePoint", "OneDrive", "Local", "WOPI", etc., and explicitly not the actual location of the file.
- **Data_UploadButtonPressed** - A Boolean to indicate if the user chose to upload the current document to a cloud location.
- **Data_UploadError** - A numerical value that indicates the kind of error that occurred if an upload operation fails.

- **Data_UpsellAppearsFromDelegate** - A Boolean value to indicate if the view was shown from the share menu.

Office.Extensibility.Catalog.ExchangeProcessEntitlement

Data regarding the processing of an individual entitlement of and Office 365 tenant admin assigned add-in.

Used in charting (requested by team management) of customer success and analysis of customer problems.

The following fields are collected:

- **AppVersion** – the version of the add-in host application
- **SolutionId** – a GUID representing a unique add-in
- **TelemetryId** – a GUID representing a unique user

Office.Extensibility.Catalog.ExchangeProcessManifest

Data regarding the processing of an individual manifest for an Office 365 tenant admin assigned add-in. Used in analysis of customer problems and charting of customer success.

The following fields are collected:

- **AppVersion** - version of the app
- **IsAllReturnedManifestsParsed** - bool indicating we parsed all returned manifests
- **IsAppCommand** - bool indicating if this is an app command app
- **ReturnedManifestsParsed** - count of the parsed manifests
- **SolutionId** - ID of the solution
- **TelemetryId** - telemetry ID based on the signed in identity

Office.Extensibility.ODPAppCommandsRibbonClick

Collects whether clicking the custom add-in control succeeded or not. Used to detect issues in user interaction with add-in controls.

The following fields are collected:

- **CommandActionType** - type of the add-in command
- **CommandLabel** - label of the command clicked
- **SolutionId** - ID of the solution

Office.Feed.Events.Initializing

This event is collected when the feed has started initializing. This event is used to indicate that the feed is starting and to diagnose reliability issues in launching the feed.

- **AppInfo.Language** - Language of the App in IETF language tag format.
- **AppInfo.Name** - Name of the component in use (Office Feed).
- **AppInfo.Version** - The version of the App.
- **clientCorrelationId** - The globally unique identifier for the application's session.
- **clientType** - The application on which the component runs.
- **DeviceInfo.Make** - The Device manufacturer or device OEM name.
- **DeviceInfo.NetworkProvider** - The network or mobile operator, such as "AT&T".

- **DeviceInfo.NetworkType** - The type of network connectivity of the device in use, such as "Wired", "Wifi" or "WWAN" (data/cellular).
- **DeviceInfo.OsName** - The name of the device OS.
- **DeviceInfo.SDKUid** - Uniquely identifies the device from the telemetry SDK's perspective.
- **eventId** - Name identifier of the event.
- **EventInfo.SdkVersion** - The version of the telemetry SDK used by the client to generate the event.
- **eventpriority** - An enumeration value for the priority of sending the event.
- **feature** - Used to group various events of the same feature.
- **hostAppRing** - The population of users to whom the application was distributed.
- **properties** - Contains additional metadata properties collected for each event.
  - **ClientTimeStamp** - Timestamp of when the event was logged in the client.
- **publicEventName** - Public facing event name.
- **region** - The geographical region of the feed service that the user is connected to.
- **tenantAadObjectId** - A globally unique identifier for the user's enterprise tenant.
- **type** - Type of the logged event, for example, Trace, Error, Event, QoS.
- **userAadObjectId** - The globally unique user identifier for an enterprise Microsoft account.
- **UserInfo.Id** - The globally unique user identifier for an enterprise Microsoft account.
- **UserInfo.IdType** - Specifies the type of a user ID.
- **UserInfo.Language** - The user's language in IETF language tag format.
- **UserInfo.MsaId** - The globally unique user identifier for a consumer Microsoft account.
- **UserInfo.OMSTenantId** - The tenant that a user's subscription is tied to. Allows us to classify issues and identify whether a problem is widespread or isolated to a set of users or a specific tenant.
- **UserInfo.TimeZone** - The user's time zone relative to UTC.
- **userPuid** - The globally unique user identifier for a consumer Microsoft account.
- **version** - The version of the feed client.

Office.Feed.Events.OfficeFeedDidAppear

This event is collected when the feed is shown to the user. The event is used to verify that the feed completed initialization step and to diagnose reliability issues in launching the feed.

- **AppInfo.Language** - Language of the App in IETF language tag format.
- **AppInfo.Name** - Name of the component in use (Office Feed).
- **AppInfo.Version** - The version of the App.
- **bridgeWaitingTime** - Metric to diagnose performance in rendering of the feed.
- **clientCorrelationId** - The globally unique identifier for the application's session.
- **clientScenario** - Scenario discriminator for different variants of the feed.
- **ClientTimeStamp** - Timestamp of when the event was logged in the client.

- **clientType** - The application on which the component runs.
- **DeviceInfo.Make** - The Device manufacturer or device OEM name.
- **DeviceInfo.NetworkProvider** - The network or mobile operator, such as "AT&T".
- **DeviceInfo.NetworkType** - The type of network connectivity of the device in use, such as "Wired", "Wifi" or "WWAN" (data/cellular).
- **DeviceInfo.OsName** - The name of the device OS.
- **DeviceInfo.SDKUid** - Uniquely identifies the device from the telemetry SDK's perspective.
- **eventId** - Name identifier of the event.
- **EventInfo.SdkVersion** - The version of the telemetry SDK used by the client to generate the event.
- **eventpriority** - An enumeration value for the priority of sending the event.
- **feature** - Used to group various events of the same feature.
- **hostAppRing** - The population of users to whom the application was distributed.
- **properties** - Contains additional metadata properties collected for each event. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **publicEventName** - Public facing event name.
- **region** - The geographical region of the feed service that the user is connected to.
- **renderTime** - Metric to diagnose performance in rendering of the feed.
- **tenantAadObjectId** - A globally unique identifier for the user's enterprise tenant.
- **type** - Type of the logged event, for example, Trace, Error, Event, QoS.
- **userAadObjectId** - The globally unique user identifier for an enterprise Microsoft account.
- **UserInfo.Id** - The globally unique user identifier for an enterprise Microsoft account.
- **UserInfo.IdType** - Specifies the type of a user ID.
- **UserInfo.Language** - The user's language in IETF language tag format.
- **UserInfo.MsaId** - The globally unique user identifier for a consumer Microsoft account.
- **UserInfo.OMSTenantId** - The tenant that a user's subscription is tied to. Allows us to classify issues and identify whether a problem is widespread or isolated to a set of users or a specific tenant.
- **UserInfo.TimeZone** - The user's time zone relative to UTC.
- **userPuid** - The globally unique user identifier for a consumer Microsoft account.
- **version** - The version of the feed client.

Office.Feedback.Survey.FloodgateClient.GetDecisionForActionPreStart

In Office apps we control the frequency of in-product and push messages through a governance layer. This event gets logged in error conditions when we try to apply governance to in-app messages before the module that is handling governance is fully activated. This telemetry helps make our governance logic more robust by collecting details of the scenarios in which the governance is not being applied.

The following fields are collected:

- **Data_EventId** - Unique identifier of the log statement.

- **Data_SurveyId** - Name of the message that we are trying to show when this error is generated.

Office.Feedback.Survey.FloodgateClient.SurveyTracked

Tracks when a device that is eligible for a survey starts an app. Used to assess the health of the survey user selection process and to ensure the signal used to analyze customer issues and health is working properly.

The following fields are collected:

- **ExpirationTimeUTC** – date/time the survey will expire
- **SurveyName** – name of survey shown
- **SurveyId** – Unique instance of a campaign
- **UniqueId** – ID to identify the individual piece of telemetry

Office.Feedback.Survey.FloodgateClient.TriggerMet

Tracks when a device has met the criteria to show a survey. Used to assess the health of the survey triggering process and to ensure the signal used to analyze customer issues and health is working properly.

The following fields are collected:

- **ExpirationTimeUTC** – date/time the survey will expire
- **SurveyName** – name of survey shown
- **SurveyId** – Unique instance of a campaign
- **UniqueId** – ID to identify the individual piece of telemetry

Office.Feedback.Survey.FloodgateClient.UserSelected

Tracks when a device has been selected for a survey. Used to assess the health of the survey user selection process and to ensure the signal used to analyze customer issues and health is working properly.

The following fields are collected:

- **ExpirationTimeUTC** – date/time the survey will expire
- **SurveyName** – name of survey shown
- **SurveyId** – Unique instance of a campaign
- **UniqueId** – ID to identify the individual piece of telemetry

Office.Feedback.Survey.UI.Android

On an Android device, it tracks when a user on a device interacts with the survey prompt and survey UI. Used to assess the health of the end-to-end survey experience and to ensure the signal used to analyze customer issues and health is working properly.

The following fields are collected:

- **ExpirationTimeUTC** – date/time the survey will expire
- **SurveyName** – name of survey shown
- **SurveyId** – Unique instance of a campaign
- **UniqueId** – ID to identify the individual piece of telemetry

Office.Feedback.Survey.UI.IOS

On an iOS device, it tracks when a user on a device interacts with the survey prompt and survey UI. Used to assess the health of the end-to-end survey experience and to ensure the signal used to analyze customer issues and health is working properly.

The following fields are collected:

- **ExpirationTimeUTC** – date/time the survey will expire
- **SurveyName** – name of survey shown
- **SurveyId** – Unique instance of a campaign
- **UniqueId** – ID to identify the individual piece of telemetry

Office.Feedback.Survey.UI.Mac

On a Mac device, it tracks when a user on a device interacts with the survey prompt and survey UI. Used to assess the health of the end-to-end survey experience and to ensure the signal used to analyze customer issues and health is working properly.

The following fields are collected:

- **ExpirationTimeUTC** – date/time the survey will expire
- **SurveyName** – name of survey shown
- **SurveyId** – Unique instance of a campaign
- **UniqueId** – ID to identify the individual piece of telemetry

Office.Feedback.Survey.UI.Win32

On a Win32 device, it tracks when a user on a device interacts with the survey prompt and survey UI. Used to assess the health of the end-to-end survey experience and to ensure the signal used to analyze customer issues and health is working properly.

The following fields are collected:

- **ExpirationTimeUTC** – date/time the survey will expire
- **SurveyName** – name of survey shown
- **SurveyId** – Unique instance of a campaign
- **UniqueId** – ID to identify the individual piece of telemetry

Office.Feedback.Survey.UI.Win32.Toast

Tracks when survey prompt is shown. Used to assess the health of the survey prompt process and to ensure the signal used to analyze customer issues and health is working properly.

The following fields are collected:

- **ExpirationTimeUTC** – date/time the survey will expire
- **SurveyName** – name of survey shown
- **SurveyId** – Unique instance of a campaign
- **UniqueId** – ID to identify the individual piece of telemetry

Office.FileIO.CSI.CCachedFileCsiLoadFileBasic

Allows us to know if a file successfully opened from the FIO Layer. Used for feature health and monitoring.

The following fields are collected:

- **Activity.Group** - tag that allows a set of monitoring events to be grouped to manage overall success
- **Activity.IsHVA** - flag to indicate that event is critical to user success
- **Data.AsyncOpen** - flag to indicate the open had content that arrived after the main body was opened

- **Data.CacheFileId** - connects to Office Document Cache telemetry to enable impact analysis of cache issues on the user experience
- **Data.CFREnabled** - Indicates that CacheFileRuntime is enabled for the session.
- **Data.CFRFailure** - Indicated that CacheFileRuntime ran into error.
- **Data.CoauthStatus** - reports collaborative status of the document on Open
- **Data.CountOfMultiRoundTripsDownload** - Count of round trips to the server used to troubleshoot performance and network issues
- **Data.CountOfMultiRoundTripsUpload** - Count of round trips to the server used to troubleshoot performance and network issues
- **Data.DialogId** - Set if a UI dialog was displayed during Open, indicating that a warning message was displayed to the user
- **Data.DidFallbackToDAV** - Set if the document was opened using an older file transfer protocol
- **Data.Doc.AccessMode** - Document is read only/editable
- **Data.Doc.AssistedReadingReasons** - Set if the document has electronic data protection in place
- **Data.Doc.AsyncOpenKind** – Indicates whether a cached version of the cloud document was opened and which asynchronous refresh logic was used.
- **Data.Doc.ChunkingType** - Units used for incremental document open
- **Data.Doc.EdpState** - Electronic Data Protection setting for the document
- **Data.Doc.Ext** - Document extension (docx/xlsb/pptx, etc.)
- **Data.Doc.Extension** - Obsolete
- **Data.Doc.FileFormat** - File format protocol version
- **Data.Doc.Fqdn** - OneDrive or SharePoint Online Domain Name
- **Data.Doc.FqdnHash** - One-way hash of customer identifiable domain name
- **Data.Doc.IdentityTelemetryId** - A one-way hash of the user identity used to perform the open
- **Data.Doc.IdentityUniqueId** - Obsolete
- **Data.Doc.InitializationScenario** - Records how the document was opened
- **Data.Doc.IOFlags** - Reports on the cached flags used to set request options
- **Data.Doc.IrmRights** - Actions permitted by the Electronic Data Protection policy that has been applied to the document/user
- **Data.Doc.IsCloudCollabEnabled** - Flag indicating that the service supports Cloud Collaboration
- **Data.Doc.IsIncrementalOpen** - Flag indicating that the document has been incrementally opened
- **Data.Doc.IsOcsSupported** - Flag indicating that the document is supported in the collaboration service
- **Data.Doc.IsOpeningOfflineCopy** - Flag indicating that the offline copy of a document was opened
- **Data.Doc.IsSyncBacked** - Flag indicating that an auto synced copy of the document exists on the computer

- **Data.Doc.Location** - Indicates which service provided the document (OneDrive, File Server, SharePoint etc.)
- **Data.Doc.LocationDetails** - Indicates which Known Folder provided a locally stored document
- **Data.Doc.NumberCoAuthors** - Count of the number of fellow users in a collaborative editing session
- **Data.Doc.PasswordFlags** - Indicates read or read/write password flags set
- **Data.Doc.ReadOnlyReasons** - Reasons why the document was opened read only
- **Data.Doc.ResourceIdHash** - An anonymized document identifier used to diagnose problems
- **Data.Doc.ServerDocId** - An immutable anonymized document identifier used to diagnose problems
- **Data.Doc.ServerProtocol** - the protocol version used to communicate with the service
- **Data.Doc.ServerType** - the type of the server offering the service (SharePoint, OneDrive, WOPI etc.)
- **Data.Doc.ServerVersion** - the server version offering the service
- **Data.Doc.SessionId** - Identifies a specific document edit session within the full session
- **Data.Doc.SharePointServiceContext** - Diagnostic information from SharePoint Online requests
- **Data.Doc.SizeInBytes** - Indicator of document size
- **Data.Doc.SpecialChars** - Indicator of special chars in the document's URL or Path
- **Data.Doc.StorageProviderId** - Obsolete
- **Data.Doc.StreamAvailability** - Indicator if document stream is available/disabled
- **Data.Doc.SyncBackedType** - Indicator as to the type of document (local or service based)
- **Data.Doc.UrlHash** - One-way hash to create a naïve document identifier
- **Data.Doc.UsedWrsDataOnOpen** - Diagnostic indicator for incremental document open
- **Data.Doc.WopiServiceId** - Contains unique identifier of WOPI service provider
- **Data.DocumentLoadEndpoint** - obsolete/redundant duplicate of (Data.Doc.Location and Data.Doc.IsSyncbacked)
- **Data.DocumentSizeInBytes** - Obsolete/redundant supplanted by Data.Doc. SizeInBytes
- **Data.DocumentSizeOnDisk** - Obsolete
- **Data.DoesBaseHaveContentOnOpen** - Change tracking diagnostic making sure we have the latest version of a shared file
- **Data.DoesWorkingBranchHaveExcludedDataOnOpen** - Change tracking diagnostic making sure we have the latest version of a shared file
- **Data.DownloadFragmentSize** - Size of data sent in a sub request for diagnosing network issues
- **Data.DsmcStartedTooEarly** - Indicates an error starting a collaborative edit session
- **Data.EditorsCount** - A count of other collaborators editing the document
- **Data.ExcludedDataThresholdInBytes** - File size required for Asynch open to be used
- **Data.FileIOResult.Code** - Cache of last Open return code from protocol layer
- **Data.FileIOResult.Success** - Cache of last Open success indicator from protocol layer

- **Data.FileIOResult.Tag** - Cache of last Open error tag from protocol layer
- **Data.FileIOResult.Type** - Cache of last Open error type from protocol layer
- **Data.FqdnHash** - Obsolete, replaced by Data_Doc_FqdnHash
- **Data.FullIError** - Cache of all Open error codes from the protocol layer
- **Data.FullyQualifiedDomainName** - Obsolete, replaced by Data_Doc_Fqdn
- **Data.Input.FileOpenState** - State requested by app (Read/ReadWrite etc.)
- **Data.Input.OpenAsync** - Async open requested by app
- **Data.Input.OpenOfflineCopy** - Open from offline copy requested by add
- **Data.IOFlags** - Obsolete
- **Data.IsBaseBranchEmptyOnOpen** - Change tracking diagnostic making sure we have the latest version of a shared file
- **Data.IsCachedHistoricalVersion** - Cache contains an older version of the document
- **Data.IsDocEnterpriseProtected** - Document has been protected by encryption (Electronic Document Protection / EDP)
- **Data.IsDocInODC** - Document has been opened before and is already in the cache
- **Data.IsMapUnMapCase** - Part of state of cached file
- **Data.IsMapUnMapCase.End** - Part of state of cached file
- **Data.IsOfficeHydrationInProgress** - The document is being restored from offline storage by Windows
- **Data.isOfficeHydrationRequired** - The document is currently in offline storage
- **Data.isOpenFromCollab** - The latest copy of the document was retrieved from the shared collaboration service
- **Data.isPendingNameExist** - Document rename is in progress
- **Data.IsStubFile** - The document hasn't been saved to the cloud service yet
- **Data.IsSyncBackedStateDifferentThanOnLastOpen** - the document state has changed, changes may have arrived while the document wasn't open
- **Data.isTaskCanceledAfterOpenComplete** - Obsolete
- **Data.IsWorkingBranchAvailableOnOpen** - Change tracking diagnostic making sure we have the latest version of a shared file
- **Data.LicenseStatus** - Diagnostic product license status, used to validate that appropriate product features are enabled for the user's license type
- **Data.LicenseType** - Indicates state of license (free/paid/trial etc.)
- **Data.Location** - Indicates storage media type/location (USB, Cloud, etc.)
- **Data.LockRequestDocMode** - Indicates if the document is available to others
- **Data.MyDeferredValue** - Obsolete
- **Data.Network.BytesReceived** - Obsolete

- **Data.Network.BytesSent** - Obsolete
- **Data.Network.ConnectionsCreated** - Obsolete
- **Data.Network.ConnectionsEnded** - Obsolete
- **Data.OcsDisableReasons** - Reason why the shared collaboration service wasn't available for the document
- **Data.OcsHostOnOpen** - Flag indicating that control will switch to the shared collaboration service during Open
- **Data.OpeningOfflineCopy** - Flag indicating that the local copy of the document will be opened
- **Data.Partition** - Obsolete
- **Data.RequestTime** - Obsolete
- **Data.ResourceIdHash** - Obsolete
- **Data.ResumedIncrementalOpen** - Obsolete
- **Data.RTCEnabled** - the fast change distribution protocol has started
- **Data.SaveOnOpen** - unsaved changes in the local document were saved to the service during Open
- **Data.ServerProtocol** - Obsolete, replaced by Data_Doc_ServerProtocol
- **Data.ServerType** - Obsolete, replaced by Data_Doc_ServerType
- **Data.ServerVersion** - Obsolete, replaced by Data_Doc_ServerVersion
- **Data.ServiceId** - Obsolete, replaced by Data_Doc_WopiServiceId
- **Data.SessionId** - Obsolete
- **Data.ShouldSwitchToServerOnly** - the local copy of the document cannot be used, and the server version must be used
- **Data.SpecialChars** - Obsolete
- **Data.StopwatchDuration** - Obsolete
- **Data.SyncBackedFileTelemetrySessionId** - Obsolete
- **Data.SyncElapsedTime** - Obsolete
- **Data.SyncRequestId** - Obsolete
- **Data.TestProperty** - Obsolete
- **Data.TransitionToHostOnOpen** - flag indicating that the session will connect to the service hosting the document
- **Data.TransitionToHostOnOpenResult** - status of the transition to the host service
- **Data.UseCachedNetworkConnection** - flag to indicate if a connection was reused or a new connection created
- **Data.UseClientIdAsSchemaLockId** - flag to control how documents are locked in the service
- **Data.VersionType** - Indicate which version type the current open operation is.
- **Data.WopiServiceId** - Obsolete, replaced by Data_Doc_WopiServiceId

Office.FileIO.CSI.CCachedFileCsiSaveFileBasic

This event allows us to know if a file was successfully saved from the FIO Layer. Used for Feature Health and monitoring.

The following fields are collected:

- **Activity.Group** - tag that allows a set of monitoring events to be grouped to manage overall success
- **Activity.IsHVA** - flag to indicate that event is critical to user success
- **Data.AsyncOpen** - flag to indicate that the document was opened with content that arrived after the main body was opened
- **Data.BaseDownloadTriggered** - Change tracking diagnostic indicating that the base version of the document was requested
- **Data.BlockAutoUploadReasons** - Reason codes for blocked upload state (for example, Autosave is turned off, the document is transitioning)
- **Data.BlockUploadDueToFailedSaveAsOverExisting** - Upload is blocked as it would fail if retried
- **Data.CacheFileId** - connects to Office Document Cache telemetry to enable impact analysis of cache issues on the user experience
- **Data.ChartType** - Obsolete
- **Data.CoAuthStatus** - reports collaborative status of the document on Save
- **Data.CoauthUpdatesContext** - reports context (Merge/Incremental Open)
- **Data.CountOfMultiRoundTripsDownload** - Count of round trips to the server used to troubleshoot performance and network issues
- **Data.CountOfMultiRoundTripsUpload** - Count of round trips to the server used to troubleshoot performance and network issues
- **Data.CFREnabled** - Indicates that CacheFileRuntime is enabled for the session.
- **Data.CFRFailure** - Indicated that CacheFileRuntime ran into error.
- **Data.DialogChoice** - Records choice made in any error dialogs
- **Data.DialogId** - Records the DialogId of any error dialogs that display during save
- **Data.Dmc.IsOcsSupported** - Obsolete
- **Data.Doc.AccessMode** - Document is read only
- **Data.Doc.AssistedReadingReasons** - Set if the document has electronic data protection in place
- **Data.Doc.AsyncOpenKind** – Indicates whether a cached version of the cloud document was opened and which asynchronous refresh logic was used.
- **Data.Doc.ChunkingType** - Units used for incremental document open
- **Data.Doc.EdpState** - Electronic Data Protection setting for the document
- **Data.Doc.Ext** - Document extension (docx/xlsm/pptx etc.)
- **Data.Doc.Extension** - Obsolete
- **Data.Doc.FileFormat** - File format protocol version
- **Data.Doc.Fqdn** - OneDrive or SharePoint Online Domain name

- **Data.Doc.FqdnHash** - One-way hash of the customer identifiable domain name
- **Data.Doc.FqdnHasi** - Obsolete
- **Data.Doc.IdentityTelemetryId** - A one-way hash of the user identity used to perform the save
- **Data.Doc.IdentityUniqueId** - Obsolete
- **Data.Doc.IKFlags** - Obsolete
- **Data.Doc.InitializationScenario** - Records how the document was opened
- **Data.Doc.IOFlags** - Reports on the cached flags used to set request options
- **Data.Doc.IrmRights** - Actions permitted by the Electronic Data Protection policy that has been applied to the document/user
- **Data.Doc.IsCloudCollabEnabled** - Flag indicating that the application supports Cloud Collaboration
- **Data.Doc.IsIncrementalOpen** - Flag indicating that the document was opened incrementally
- **Data.Doc.IsOcsSupported** - Flag indicating that the document supports Cloud Collaboration
- **Data.Doc.IsOpeningOfflineCopy** - Flag indicating that the offline copy of a document was opened
- **Data.Doc.IsSyncBacked** - Flag indicating that an auto synced copy of the document exists on the computer
- **Data.Doc.Location** - Indicates which service provided the document (OneDrive, File Server, SharePoint etc.)
- **Data.Doc.LocationDetails** - Indicates which Known Folder provided a locally stored document
- **Data.Doc.NumberCoAuthors** - Count of the number of fellow users in a collaborative editing session
- **Data.Doc.PasswordFlags** - Indicates read or read/write password flags set
- **Data.Doc.ReadOnlyReasons** - Reasons why the document was opened read only
- **Data.Doc.ResourceIdHash** - An anonymized document identifier used to diagnose problems
- **Data.Doc.ServerDocId** - An immutable anonymized document identifier used to diagnose problems
- **Data.Doc.ServerProtocol** - the protocol version used to communicate with the service
- **Data.Doc.ServerType** - the type of the server offering the service (SharePoint, OneDrive, WOPI etc.)
- **Data.Doc.ServerVersion** - the server version offering the service
- **Data.Doc.SessionId** - Identifies a specific document edit session within the full session
- **Data.Doc.SharePointServiceContext** - Diagnostic information from SharePoint Online requests
- **Data.Doc.SizeInBytes** - Indicator of document size
- **Data.Doc.SpecialChars** - Indicator of special chars in the document's URL or Path
- **Data.Doc.StorageProviderId** - Obsolete
- **Data.Doc.StreamAvailability** - Indicator if document stream is available/disabled
- **Data.Doc.SussionId** - Obsolete
- **Data.Doc.SyncBackedType** - Indicator as to the type of document (local or service based)
- **Data.Doc.UrlHash** - One-way hash to create a naïve document identifier

- **Data.Doc.UsedWrsDataOnOpen** - Diagnostic indicator for incremental document open
- **Data.Doc.WopiServiceId** - Contains unique identifier of WOPI service provider
- **Data.DocnReadOnlyReasons** - Obsolete
- **Data.DocumentSaveEndpoint** - Obsolete, replaced by Data_Doc_Location
- **Data.DocumentSaveType** - Type of Save (Normal, Create, SaveAs)
- **Data.DocumentSizeOnDisk** - Obsolete, replaced by Data_Doc_SizeInBytes
- **Data.DoesBaseHaveContentOnOpen** - Change tracking diagnostic making sure we have the latest version of a shared file
- **Data.DoesWorkingBranchHaveExcludedDataOnOpen** - Change tracking diagnostic making sure we have the latest version of a shared file
- **Data.DstDoc.AccessMode** - New document is read only/editable
- **Data.DstDoc.EdpState** - Electronic Data Protection setting for the new document
- **Data.DstDoc.Extension** - New document's extension (docx/xlsm/pptx, etc.)
- **Data.DstDoc.FileFormat** - New document's file format protocol
- **Data.DstDoc.Fqdn** - New document's OneDrive or SharePoint Online domain name
- **Data.DstDoc.FqdnHash** - One-way hash of new document's customer identifiable domain name
- **Data.DstDoc.IdentityUniqueId** - Obsolete
- **Data.DstDoc.IOFlags** - New document's cached options flags used when opening
- **Data.DstDoc.IsOpeningOfflineCopy** - Flag indicating that an offline copy of the new document was opened
- **Data.DstDoc.IsSyncBacked** - Flag indicating that an auto synced copy of the document exists on the computer
- **Data.DstDoc.Location** - Indicates which service provided the new document (OneDrive, File Server, SharePoint, etc.)
- **Data.DstDoc.NumberCoAuthors** - Count of the number of fellow users in a collaborative editing session on the new document
- **Data.DstDoc.ReadOnlyReasons** - Reasons why the new document was opened read only
- **Data.DstDoc.ResourceIdHash** - An anonymized document identifier used to diagnose problems with the new document
- **Data.DstDoc.ServerDocId** - An immutable anonymized document identifier used to diagnose problems with the new document
- **Data.DstDoc.ServerProtocol** - the protocol version used to communicate with the service when creating the new document
- **Data.DstDoc.ServerType** - the type of the server offering the service (SharePoint, OneDrive, WOPI etc.) for the new document
- **Data.DstDoc.ServerVersion** - the server version offering the service for the new document
- **Data.DstDoc.SessionId** - Identifies a specific document edit session within the full session for the new

document

- **Data.DstDoc.SharePointServiceContext** - Diagnostic information from SharePoint Online requests for the new document
- **Data.DstDoc.SizeInBytes** - Indicator of document size of new document
- **Data.DstDoc.UrlHash** - One-way hash to create a naïve document identifier for the new document
- **Data.EditorsCount** - A count of other collaborators editing the document
- **Data.FullIError** - Cache of all error codes from the protocol layer
- **Data.HasFilteredCategories** - Obsolete
- **Data.HasFilteredCategoryNames** - Obsolete
- **Data.HasFilteredSeries** - Obsolete
- **Data.HasFilteredSeriesNames** - Obsolete
- **Data.HasPendingSaveAs** - Indicates that a request Save As/Save a Copy is in progress
- **Data.Input.FileOpenState** - State requested by app (Read/ReadWrite, etc.)
- **Data.Input.FileSaveState** - State requested by app (Save on Open, Save As, etc.)
- **Data.Input.NetworkCost** - Indicates network cost/type (metered, metered above cap, etc.)
- **Data.Input.OpenAsync** - Flag indicates app requested an async open
- **Data.Input.OpenOfflineCopy** - Flag indicates app requested an offline open
- **Data.IsCachedHistoricalVersion** - Indicates that this cached file is not the latest version
- **Data.IsHtml** - Indicates that HTML format text was pasted
- **Data.IsLegacyCode** - Indicates that Legacy code format text was pasted
- **Data.IsLocalOnlyFile** - Indicates that the file was opened from local storage only
- **Data.IsLocalOrSyncBackedFile** - Indicates that the file was opened locally and mapped through to the service
- **Data.IsMapUnMapCase** - Part of state of cached file
- **Data.isOpenFromCollab** - Indicates that the file was opened from the shared collaboration service
- **Data.IsStubFile** - The document hasn't been shared to the cloud service yet
- **Data.IsSyncBackedFile** - the document is in a folder that is auto sync updated
- **Data.IsSyncBackedStateDifferentThanOnLastOpen** - the document state has changed, changes may have arrived while the document wasn't open
- **Data.IsWorkingBranchAvailableOnOpen** - change tracking diagnostic making sure we have the latest version of a shared file
- **Data.Location** - Indicates storage media type/location (USB; Cloud, etc.)
- **Data.LockRequestDocMode** - Indicates if the document is available to others
- **Data.MruRequestResult** - Obsolete
- **Data.NewDataNotAvailableReason** - Obsolete

- **Data.OcsDisableReasons** - Not used by Save
- **Data.OcsHostOnOpen** - Not used by Save
- **Data.Output.FileSaveState** - State on save completion
- **Data.PivotChart** - Obsolete
- **Data.resolveConflictState** - Reason codes for a request to resolve merge conflicts
- **Data.RTCEnabled** - the fast change distribution protocol has started
- **Data.SaveAsToCurrent** - Indicates that the active document will overwrite the stored file
- **Data.ServiceId** - Obsolete, replaced by Data_Doc_WopiServiceId
- **Data.SessionId** - Obsolete
- **Data.SizeInBytes** - Obsolete, replaced by Data_Doc_SizeInBytes
- **Data.StopwatchDuration** - Obsolete
- **Data.SyncBackedFileRequiresOnlineTransition** - Flag indicating that Save action is temporarily blocked by online transition
- **Data.SyncBackedFileSaveOnOpen** - Flag indicating that changes made by auto sync require a save on open
- **Data.TelemetryId** - Obsolete
- **Data.TriggerSaveAfterBaseDownload** - change tracking diagnostic making sure we have the latest version of a shared file
- **Data.UploadBlockedDueToCoherencyFailure** - Save to service blocked pending user resolution of conflicting changes
- **Data.UploadBlockedDueToFailedSaveAsOverExisting** - Save to service blocked due to failed attempt to overwrite an existing file
- **Data.UploadPreemptedForCoherency** - Save to service abandoned as more changes are being made by the user
- **Data.UploadPreemptedForSaveAsOverExistingFailure** - Save to service abandoned due to earlier SaveAsOverExisting failure
- **Data.UploadScheduled** - file is ready to be asynchronously uploaded to the service
- **Data.UseClientIdAsSchemaLockId** - flag to control how documents are locked in the service
- **Data.WorkingCopySaved** - change tracking diagnostic making sure we have the latest version of a shared file
- **Data.ZrtSaveAsforSyncBackedBusinessEnabled** - flag indicating fast save enabled for SharePoint Online
- **Data.ZrtSaveAsforSyncBackedConsumerEnabled** - flag indicating fast save enabled for OneDrive Consumer
- **Data.ZrtSaveAsforSyncBackedCTBusinessEnabled** - flag indicating fast save content types enabled for SharePoint Online
- **Data.ZrtSaveAsforSyncBackedCTConsumerEnabled** - flag indicating fast save content types enabled for OneDrive Consumer

- **Data.ZrtSaveAsforSyncBackedMetaDataBusinessEnabled** - flag indicating fast file metadata save enabled for SharePoint Online
- **Data.ZrtSaveAsforSyncBackedMetaDataConsumerEnabled** - flag indicating fast file metadata save enabled for OneDrive Consumer-

Office.FindTime.AppFailedToStart

Collected when app fails to start due to an unexpected error during startup. Used to track exceptions & crashes. Helps monitor & debug app health.

The following fields are collected:

- **DateTime** - Timestamp of when the event is logged
- **EventName** - The name of the event being logged

Office.FirstRun.Apple.ActivationResult

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of our application activation flow. We collect data to figure out the outcome of the Office 365 subscription activation along with the flow used to activate (First Run Experience, In-App-Flow, Purchase, etc.).

The following fields are collected:

- **Data_ActivationStatusCollectionTime** – A timestamp
- **Data_ActivationStatusError** – An activation error code.
- **Data_ActivationStatusFlowType** – A numeric value indicating the type of activation flow

Office.FirstRun.Apple.ActivationStatus

This event is collected for Office applications running under Apple platforms. The event is used to figure out the outcome of the Office 365 subscription activation along with the flow used to activate (FRE, InApp, Purchase, etc.). We collect data containing the Activation type, flow type (FRE/DocStage/Purchase) and Office Licensing Service ID.

The following fields are collected:

- **Data_ActivationTypeCollectionTime** – A timestamp
- **Data_ActivationTypeFlowType** – A numeric value indicating the type of activation flow
- **Data_ActivationTypeOLSLicense** – An identifier of the License
- **Data_ActivationTypeStatus** – An activation status code.

Office.FirstRun.Apple.FirstRunComplete

This event is collected for Office applications running under Apple platforms. The event lets us know if the user running in freemium, the flow type being run (FRE/DocStage/Purchase) and the identity type (MSA/OrgID). We use this event to figure out if the First Run-Experience (FRE) was completed and type of identity used to sign in (MSA/OrgID).

The following fields are collected:

- **Data_FirstRunCompletedCollectionTime** - A timestamp registering the time at which the flow was completed
- **Data_FirstRunCompletedFlowType** - A code denoting the type of user flow that was completed
- **Data_FirstRunCompletedFreemiumStatus** - A code representing the status of completion for a freemium user flow
- **Data_FirstRunCompletedIdentityType** - The type of identity of the user that completed the flow

Office.FirstRun.Apple.FirstRunStart

This event is collected for Office applications running under Apple platforms. The event lets us know a user has entered first run experience and the flow type being run (FRE/DocStage/Purchase). We use this event to figure out if the First Run-Experience (FRE) was started successfully.

The following fields are collected:

- **Data_FirstRunStartedCollectionTime** - A timestamp registering the time at which the flow was completed
- **Data_FirstRunStartedFlowType** - A code denoting the type of user flow that was completed

Office.FirstRun.Apple.FirstRunStartedAndCompleted

This event is collected for Office applications running under Apple platforms. The event lets us know if the user running in freemium, the flow type being run (FRE/DocStage/Purchase) and the identity type (MSA/OrgID). We use this event to figure out the health and effectiveness of our First-Run Experience (FRE) flow.

The following fields are collected:

- **Data_FirstRunCompletedCollectionTime** - A timestamp registering the time at which the flow was completed
- **Data_FirstRunCompletedFlowType** - A code denoting the type of user flow that was completed
- **Data_FirstRunCompletedFreemiumStatus** - A code representing the status of completion for a freemium user flow
- **Data_FirstRunCompletedIdentityType** - The type of identity of the user that completed the flow
- **Data_FirstRunStartedCollectionTime** - A timestamp registering the time at which the flow was started
- **Data_FirstRunStartedFlowType** - A code denoting the type of user flow that was started

Office.FirstRun.Apple.InAppPurchaseActivationFail

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of our application activation flow. We collect data to figure out the outcome of the In-App purchase activation along with the flow used to activate (First Run Experience, In-App-Flow, Purchase, etc.).

The following fields are collected:

- **Data_ActivationFailCollectionTime** - A timestamp registering the time at which the activation failure occurred
- **Data_ActivationFailFlowType** - A code denoting the type of user flow that was exercised
- **Data_AssoicatedSuccessfullyCollectionTime** - A timestamp registering the time at which the association occurred
- **Data_AssoicatedSuccessfullyFlowType** - A code denoting the type of user flow that was exercised

Office.FirstRun.Apple.InAppPurchaseActivationSuccess

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of our application activation flow. We collect data to figure out the outcome of the In-App purchase activation along with the flow used to activate (First Run Experience, In-App-Flow, Purchase, etc.).

The following fields are collected:

- **Data_ActivatedSuccessfullyCollectionTime** - A timestamp registering the time at which the activation occurred

- **Data_ActivatedSuccessfullyFlowType** - A code denoting the type of user flow that was exercised
- **Data_AssoicatedSuccessfullyCollectionTime** - A timestamp registering the time at which the association occurred
- **Data_AssoicatedSuccessfullyFlowType** - A code denoting the type of user flow that was exercised

Office.FirstRun.Apple.InAppPurchaseAssociationFailed

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of our application activation flow. We collect data to figure out the outcome of the In-App purchase activation along with the flow used to activate (First Run Experience, In-App-Flow, Purchase, etc.).

The following fields are collected:

- **Data_AppChargedSuccessfullyCollectionTime** - A timestamp registering the time at which the purchase was charged
- **Data_AppChargedSuccessfullyFlowType** - A code denoting the type of user flow that was exercised
- **Data_AssoicationFailedCollectionTime** - A timestamp registering the time at which the app association failed
- **Data_AssoicationFailedFlowType** - A code denoting the type of user flow that was exercised
- **Data_AssoicationFailedResult** - A code denoting type of failure observed

Office.FirstRun.Apple.InAppPurchaseAssociationSuccess

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of our application activation flow. We collect data to figure out the outcome of the In-App purchase activation along with the flow used to activate (First Run Experience, In-App-Flow, Purchase, etc.).

The following fields are collected:

- **Data_AppChargedSuccessfullyCollectionTime** - A timestamp registering the time at which the purchase was charged
- **Data_AppChargedSuccessfullyFlowType** - A code denoting the type of user flow that was exercised
- **Data_AssoicatedSuccessfullyCollectionTime** - A timestamp registering the time at which the app association failed
- **Data_AssoicatedSuccessfullyFlowType** - A code denoting the type of user flow that was exercised

Office.FirstRun.Apple.InAppPurchaseFailures

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of our application activation flow. We collect data on the outcome of the In-App purchase flow.

The following fields are collected:

- **Data_AppStoreFailureFlowType** - A code denoting the type of user flow that was exercised
- **Data_AppStoreFailureResult** - The failure result observed
- **Data_CancelRequestFlowType** - A code denoting the type of user flow that was exercised
- **Data_EventId** - A code denoting type of failure observed

Office.FirstRun.Apple.InAppPurchasesAttempted

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of our application in-app purchase flow. We collect data to track the attempted In-App purchases and their Type of SKU being purchased (Monthly/Annual/Home/Personal).

The following fields are collected:

- **Data_EventId** - A code denoting type of result observed
- **Data_PurchasedClickedOfferType** - The type of SKU attempted to purchase
- **Data_PurchaseSuccessfulFlowType** - A code denoting the type of user flow that was exercised

Office.FirstRun.Apple.InAppRestoreAttempted

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of our application in-app purchase flow. We collect data to track the attempted In-App restorations

The following fields are collected:

- **Data_EventId** - A code denoting the type of outcome of the attempt
- **Data_RestoreAttemptFlowType** - A code denoting the type of user flow that was exercised

Office.FirstRun.Apple.InAppRestoreAttemptFailed

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of our application in-app purchase flow. We collect data to track the attempted In-App restorations and their associated flows and errors.

The following fields are collected:

- **Data_RestoreButtonFlowType** - A code denoting the type of user flow that was exercised
- **Data_RestoredFailedPaymentCancelledFlowType** - A code denoting the type of payment cancellation flow that was exercised
- **Data_RestoredFailedUnKnownFlowType** - Whether the attempt failed due to the exercise of an unexpected user flow
- **Data_RestoredFailedUnKnownResult** - Whether the attempt failed due to unknown reasons

Office.FirstRun.Apple.MacFirstRunCompleted

This event is collected for Office applications running under Apple platforms. The event lets us know a user has gone through first run experience. We use this event to figure out if the First Run-Experience (FRE) was completed successfully.

The following fields are collected:

- **Data_FirstRunCollectionTime** - A timestamp registering the time at which the flow was completed.

Office.FirstRun.Apple.MacWXPFirstRunStarted

This event is collected for Office applications running under Apple platforms. The event lets us know a user has entered first run experience. We use this event to figure out if the First Run-Experience (FRE) was started successfully.

The following fields are collected:

- **Data_FirstRunPanelName** - The name of the panel from which the experience started

Office.Floodgate.UserFact.AppUsage

This indicates when a user has used high value features within the product. It may indicate if the user discovered the feature or used it. The signal will feed feature usage product insights that help make the product better.

The following fields are collected:

- **FeatureAction** - A label indicating the high value feature and action performed by the user, for example, ContentPickerTried, TemplatesSeen.

Office.Lens.LensSdk.CloudConnectorLaunch

When the user crops the image and taps confirm on the final image selection for using OCR, this event is collected. This is user-to-request record for the service as there is no user-to-service-job mapping on the service. UserId is required to fulfill GDPR requirements as service is not directly exposed to users, but through clients and identify the total number of people using the service, helping the service track the volume of users using the product, as well as identifying changes in trends, help look for and rectify issues in the product.

The following fields are collected:

- **CallType** - String to identify if the API call was synchronous or asynchronous.
- **CloudConnectorRequestId** - String that identifies the service request that was made to convert images via service.
- **CloudConnectorTarget** - String that confirms which type of conversion the service will do on images, like converting to PDF, Docx, text, etc.
- **CustomerId** - String that identifies the user who owns the images being processed.
- **CustomerType** - String that identifies the customer as an enterprise or individual user. This distinction affects the number of images (quota) the client can convert at a time.
- **RelationId** - String, which identifies correlation between Lens and the service used to process files.

Office.Lens.LensSdk.CloudConnectorUploadError

In Image to Table, when the user taps either Share, Copy or Open, the corrections in the table made by the user are shared with the service to improve the OCR. This event is collected on the error response of that service and contains the relevant identifiers to troubleshoot various issues on the service.

The following fields are collected:

- **CloudConnectorRequestId** - String identifier to link the service job to the current service request for which the improvement data was shared.
- **CorrelationId** - String that contains the identifier of the current service job instance.
- **Reason** - String that contains the error code and description of the error.
- **TargetType** - String that identifies the endpoint on the service.
- **TaskType** - String that identifies the intent of the service call.

Office.Lens.LensSdk.CloudConnectorUploadSuccess

In Image to Table, when the user taps either Share, Copy or Open, the corrections in the table made by the user are shared with the service to improve the OCR. This event is collected on the successful response of that service and contains the relevant identifiers to troubleshoot the process. It also helps analyze the usage of service improvement pipeline.

The following fields are collected:

- **CloudConnectorRequestId** - String identifier to link the service job to the current service request for which the improvement data was shared.
- **CorrelationId** - String that contains the identifier of the current service job instance.
- **TargetType** - String that identifies the endpoint on the service.
- **TaskType** - String that identifies the intent of the service call.

Office.Lens.LensSdk.SaveMedia

This event is invoked when the user clicks on the done button and saves images on Android and iOS. It helps

measure the level of user engagement by quantifying users who end up saving images through our app.

The following fields are collected on Android:

- **Data_FileSizeAfterCleanUp** - Size of the file after it is cleaned by app, to understand how much compression was achieved after cleanup.
- **Data_FileSizeAfterSave** - Size of the file after it is saved by user, to understand how much compression was achieved after saving.
- **Data_FileSizeBeforeCleanUp** - Size of the file before it is cleaned by app, to understand how much captured size was
- **Data_Filter** - The filter applied to the image.
- **Data_ImageHeightAfterCleanUp** - Height of the image after it was cleaned by app.
- **Data_ImageHeightBeforeCleanUp** - Height of the image before it was cleaned by app.
- **Data_ImageWidthAfterCleanUp** - Width of the image before it was cleaned by app.
- **Data_ImageWidthBeforeCleanUp** - Width of the image before it was cleaned by app.
- **Data_MediaId** - Identifier for images to help track operation success.
- **Data_ProcessMode** - Mode of the user at the time of the saving the image by the user.
- **Data_Source** - Defines where the image was sourced from, example captured via camera, imported from gallery, etc.

The following fields are collected on iOS:

- **Data_filter** - The filter applied to the image.
- **Data_imageDPI** - Image reduction applied to the saved file image
- **Data_imageSize** - Size of the image after user has saved the image
- **Data_mediaId** - Identifier for images to help track operation success.
- **Data_mode** - Mode of the user at the time of the saving the image by the user.
- **Data_sizeinPixel** - Size of the image in the form of pixel
- **Data_source** - Defines where the image was sourced from, example captured via camera, imported from gallery, etc.

Office.Lens.LensSdk.ServiceIDMapping

This event is collected when Lens SDK interacts with Microsoft's Image-to-document (or I2D) service. This means that the event is called:

- When an image is uploaded to our I2D service for file conversion and extraction (OCR).
- When the user needs to correct the service's output, we send feedback to improve quality.

The data is used to analyse the usage and troubleshoot issues on the service side.

The following fields are collected:

- **CloudConnectorRequestId** - String that identifies the service-requests on the client app for both conversion and feedback scenarios.
- **CustomerId** - This string helps map users to service requests and help us track usage. UserId is required to fulfil GDPR requirements as service is not directly exposed to users, but through clients and identify

the total number of people using the service, helping the service track the volume of users using the product.

- **I2DFeedbackAPICorrelationId** - String that identifies the feedback-request in I2D service when user corrects the service output.
- **I2DServiceProcessID** - String that identifies the service-request in I2D service when user is uploading images for conversion.

Office.LivePersonaCard.ConfigurationSetAction

We log when the user is in an app that loads a Persona Card in anticipation of the user opening the Live Persona Card. The data is used to determine whether the card loaded correctly.

The following fields are collected:

- **Data.accountType** - Whether the user belongs to an organization or a consumer
- **Data.appContextId** - A randomly generated ID used to identify different accounts in the same app
- **Data.AppInfo.Name** - Name of the service in use (Profile card)
- **Data.AppInfo_Id** - Name of the host application
- **Data.AppInfo_Version** - Version of the host application
- **Data.cardCorrelationId** - The globally unique identifier for a persona card
- **Data.cardPersonaCorrelationId** - The globally unique identifier for a specific persona shown in a card
- **Data.clientCorrelationId** - The globally unique identifier for the app's session
- **Data.clientType** - The type of device the app is run on
- **Data.contextType** - What context (app) the card was launched from
- **Data.ecsConfigIds** - Version identifiers for the features enabled in the card
- **Data.ecsTagId** - Tag ID for features
- **Data.eventId** - Name identifier of the event, for example "LivePersonaCardRenderedAction"
- **Data.eventpriority** - An enumeration value for the priority of sending the event.
- **Data.feature** - Used to group various events of the same feature (Profile card)
- **Data.flights** - The features enabled in the card
- **Data.fromCache** - Whether data was fetched from memory
- **Data.hasFinePointer** - Whether the device has mouse-pointer capability
- **Data.hasHoverEvents** - Whether the device has mouse-hover capability
- **Data.immersiveProfileCorrelationId** - A globally unique identifier for the expanded profile view session
- **Data.offlineResolved** - Whether data was fetched while offline
- **Data.OTelJS.Version** - Version of OTel logger
- **Data.personaCorrelationId** - A globally unique identifier for unique personas in a session
- **Data.properties** - Additional metadata collected for each event as follows: *[This field has been removed from current builds of Office, but might still appear in older builds.]*

    - **cardCorrelationId** - Duplicate of Data.appContextId above
    - **cardPersonaCorrelationId** - Duplicate of Data.cardCorrelationId above
    - **ClientTimeStamp** - Time on the application when the event was logged
    - **consumerCorrelationId** - Duplicate of Data.clientCorrelationId above
    - **externalAppSessionCorrelationId** - A globally unique identifier for the app to identify all persona cards opened in the same sub-session
- **Data.region** -The geographical region of the profile card backend service to which user is connected
- **Data.tenantAadObjectId** - The tenant to which a user's subscription is tied. Allows us to classify issues and identify whether a problem is widespread or isolated to a set of users or a specific tenant
- **Data.type** -Type of the logged event, for example, Trace, Error, Event
- **Data.userAadObjectId** -The globally unique user identifier for an enterprise Microsoft account (duplicate of Data.UserInfo.Id)
- **Data.UserInfo.Id** - The globally unique user identifier for an enterprise Microsoft account
- **Data.UserInfo.MsaId** - The globally unique user identifier for a consumer Microsoft account
- **Data.UserInfo.OMSTenantId** - The tenant that a user's subscription is tied to. Allows us to classify issues and identify whether a problem is widespread or isolated to a set of users or a specific tenant
- **Data.userPuid** - The globally unique user identifier for a consumer Microsoft account (duplicate of Data.UserInfo.MsaId)
- **Data.version** - The version of the service (Profile Card)
- **Data.workloadCulture** - Culture set in the host application
- **DeviceInfo_Id** - The globally unique device identifier for a device
- **DeviceInfo_Make** - The brand of the operating system
- **DeviceInfo_Model** - The model of the device
- **DeviceInfo_OsName** - The name of the device OS
- **DeviceInfo_OsVersion** - The version of the operating system
- **DeviceInfo_SDKUid** - Uniquely identifies the device from the telemetry SDK's perspective

Office.LivePersonaCard.UserActions.ClosedExpandedPersonaCard

Logged when the user closes an expanded Persona Card. It is used to observe critical anomalies in failure rates of closing the Live Persona Card.

The following fields are collected:

- **AppInfo_Id** – Name of the host application
- **AppInfo_Version** – Version of the host application
- **Data.appContextId** - A randomly generated ID used to identify different accounts in the same app
- **Data.AppInfo.Name** - Name of the service in use (Profile card)
- **Data.cardCorrelationId** - The globally unique identifier for a persona card
- **Data.cardPersonaCorrelationId** - The globally unique identifier for a specific persona shown in a card

- **Data.clientCorrelationId** - The globally unique identifier for the app's session
- **Data.clientType** - The type of device the app is run on, for example “Outlook_Win32”
- **Data.eventId** - Name identifier of the event, for example "LivePersonaCardRenderedAction"
- **Data.exportName** - Human readable name of the user action event, for example "ClosedExpandedPersonaCard"
- **Data.exportType** - Category of the event for GDPR export request
- **Data.externalAppSessionCorrelationId** - A globally unique identifier for the app to identify all persona cards opened in the same sub-session
- **Data.feature** - Used to group various events of the same feature (Profile card)
- **Data.immersiveProfileCorrelationId** - A globally unique identifier for the expanded profile view session
- **Data.OTelJS.Version** - Version of OTel logger
- **Data.personaCorrelationId** - A globally unique identifier for unique personas in a session
- **Data.properties** - Additional metadata collected for each event as follows: *[This field has been removed from current builds of Office, but might still appear in older builds.]*
  - **cardCorrelationId** - Duplicate of Data.appContextId above
  - **cardPersonaCorrelationId** - Duplicate of Data.cardCorrelationId above
  - **ClientTimeStamp** - time that the event occurred in Unix epoch time
  - **consumerCorrelationId** - Duplicate of Data.clientCorrelationId above
- **Data.region** -The geographical region of the profile card backend service to which user is connected
- **Data.tenantAadObjectId** - The tenant to which a user’s subscription is tied. Allows us to classify issues and identify whether a problem is widespread or isolated to a set of users or a specific tenant
- **Data.type** -Type of the logged event, for example, Trace, Error, Event
- **Data.userAadObjectId** -The globally unique user identifier for an enterprise Microsoft account (duplicate of Data.UserInfo.Id)
- **Data.UserInfo.Id** - The globally unique user identifier for an enterprise Microsoft account
- **Data.UserInfo.MsaId** - The globally unique user identifier for a consumer Microsoft account
- **Data.UserInfo.OMSTenantId** - The tenant that a user’s subscription is tied to. Allows us to classify issues and identify whether a problem is widespread or isolated to a set of users or a specific tenant.
- **Data.userPuid** -The globally unique user identifier for a consumer Microsoft account (duplicate of Data.UserInfo.MsaId)
- **Data.version** -The version of the service (Profile Card)
- **DeviceInfo_Id** – The globally unique device identifier for a device
- **DeviceInfo_Make** – The brand of the operating system
- **DeviceInfo_Model** – The model of the device
- **DeviceInfo.NetworkCost** - Indicates network cost/type (metered, metered above cap, etc.)
- **DeviceInfo_OsName** - The name of the device OS

- **DeviceInfo_OsVersion** – The version of the operating system
- **PipelineInfo.ClientCountry** - The Country Code of the Sender, based on the unscrubbed Client IP Address

Office.LivePersonaCard.UserActions.ClosedPersonaCard

We log when the user closes a Persona Card. The data is used to determine whether the card closed correctly.

The following fields are collected:

- **BatchId** - Globally unique identifier if a set of requests was made
- **Data.appContextId** - A randomly generated ID used to identify different accounts in the same app
- **Data.AppInfo.Name** - Name of the service in use (Profile card)
- **Data.AppInfo_Id** - Name of the host application
- **Data.AppInfo_Version** - Version of the host application
- **Data.cardCorrelationId** - The globally unique identifier for a persona card
- **Data.cardPersonaCorrelationId** - The globally unique identifier for a specific persona shown in a card
- **Data.clientCorrelationId** - The globally unique identifier for the app's session
- **Data.clientType** - The type of device the app is run on
- **Data.eventId** - Name identifier of the event, for example "LivePersonaCardRenderedAction"
- **Data.externalAppSessionCorrelationId** - A globally unique identifier for the app to identify all persona cards opened in the same sub-session.
- **Data.feature** - Used to group various events of the same feature (Profile card)
- **Data.immersiveProfileCorrelationId** - A globally unique identifier for the expanded profile view session
- **Data.OTelJS.Version** - Version of OTel logger
- **Data.personaCorrelationId** - A globally unique identifier for unique personas in a session
- **Data.properties** - Additional metadata collected for each event as follows: *[This field has been removed from current builds of Office, but might still appear in older builds.]*
  - **ClientTimeStamp** - Time on the application when the event was logged
  - **cardCorrelationId** - Duplicate of Data.appContextId above
  - **cardPersonaCorrelationId** - Duplicate of Data.cardCorrelationId above
  - **consumerCorrelationId** - Duplicate of Data.clientCorrelationId above
- **Data.region** -The geographical region of the profile card backend service to which user is connected
- **Data.tenantAadObjectId** - The tenant to which a user's subscription is tied. Allows us to classify issues and identify whether a problem is widespread or isolated to a set of users or a specific tenant
- **Data.type** -Type of the logged event, for example, Trace, Error, Event
- **Data.userAadObjectId** -The globally unique user identifier for an enterprise Microsoft account (duplicate of Data.UserInfo.Id)
- **Data.UserInfo.Id** - The globally unique user identifier for an enterprise Microsoft account
- **Data.UserInfo.MsaId** - The globally unique user identifier for a consumer Microsoft account

- **Data.UserInfo.OMSTenantId** - The tenant that a user's subscription is tied to. Allows us to classify issues and identify whether a problem is widespread or isolated to a set of users or a specific tenant
- **Data.userPuid** -The globally unique user identifier for a consumer Microsoft account (duplicate of Data.UserInfo.MsaId)
- **Data.version** -The version of the service (Profile Card)
- **Data_hostAppRing** - The rollout ring of the persona card
- **Event_ReceivedTime** - The time the event was logged in the service

Office.LivePersonaCard.UserActions.OpenedExpandedPersonaCard

Logged when the user opens an expanded Persona Card. It is used to observe critical anomalies in failure rates of launching the Live Persona Card.

The following fields are collected:

- **AppInfo_Id** – Name of the host application
- **AppInfo_Version** – Version of the host application
- **Data.appContextId** - A randomly generated ID used to identify different accounts in the same app
- **Data.AppInfo.Name** - Name of the service in use (Profile card)
- **Data.cardCorrelationId** - The globally unique identifier for a persona card
- **Data.cardPersonaCorrelationId** - The globally unique identifier for a specific persona shown in a card
- **Data.clientCorrelationId** - The globally unique identifier for the app's session
- **Data.clientScenario** - To identify the feature in the app from where the persona card was opened
- **Data.clientType** - The type of device the app is run on
- **Data.eventId** - Name identifier of the event, for example "LivePersonaCardRenderedAction"
- **Data.externalAppSessionCorrelationId** - A globally unique identifier for the app to identify all persona cards opened in the same sub-session.
- **Data.exportName** - Human readable name of the user action event, for example "OpenedPersonaCard"
- **Data.exportType** - Category of the event for GDPR export request
- **Data.feature** - Used to group various events of the same feature (Profile card)
- **Data.hasPersonaInsightRing** - Insights from Office or LinkedIn could be available for the user
- **Data.hostAppRing** - The ring by which the app was distributed
- **Data.immersiveProfileCorrelationId** - A globally unique identifier for the expanded profile view session
- **Data.OTelJS.Version** - Version of OTel logger
- **Data.personaCorrelationId** - A globally unique identifier for unique personas in a session
- **Data.properties** - Additional metadata collected for each event as follows: *[This field has been removed from current builds of Office, but might still appear in older builds.]*
  - **cardCorrelationId** - Duplicate of Data.appContextId above
  - **cardPersonaCorrelationId** - Duplicate of Data.cardCorrelationId above
  - **consumerCorrelationId** - Duplicate of Data.clientCorrelationId above

- **Data.region** -The geographical region of the profile card backend service to which user is connected
- **Data.section** – The active section of the expanded card
- **Data.tenantAadObjectId** - The tenant to which a user's subscription is tied. Allows us to classify issues and identify whether a problem is widespread or isolated to a set of users or a specific tenant
- **Data.type** -Type of the logged event, for example, Trace, Error, Event
- **Data.userAadObjectId** -The globally unique user identifier for an enterprise Microsoft account (duplicate of Data.UserInfo.Id)
- **Data.UserInfo.Id** - The globally unique user identifier for an enterprise Microsoft account
- **Data.UserInfo.MsaId** - The globally unique user identifier for a consumer Microsoft account
- **Data.UserInfo.OMSTenantId** - The tenant that a user's subscription is tied to. Allows us to classify issues and identify whether a problem is widespread or isolated to a set of users or a specific tenant
- **Data.userPuid** -The globally unique user identifier for a consumer Microsoft account (duplicate of Data.UserInfo.MsaId)
- **Data.version** -The version of the service (Profile Card)
- **DeviceInfo_Id** – The globally unique device identifier for a device
- **DeviceInfo_Make** – The brand of the operating system
- **DeviceInfo_Model** – The model of the device
- **DeviceInfo_OsName** - The name of the device OS
- **DeviceInfo_OsVersion** – The version of the operating system
- **DeviceInfo_SDKUid** – Uniquely identifies the device from the telemetry SDK's perspective
- **NetworkCost** - Indicates network cost/type (metered, metered above cap, etc.)
- **NetworkCountry** - The Country Code of the Sender, based on the unscrubbed Client IP Address

Office.LivePersonaCard.UserActions.OpenedPersonaCard

Logged when the user opens a Persona Card. It is used to observe critical anomalies in failure rates of launching the Live Persona Card.

The following fields are collected:

- **Data.appContextId** - A randomly generated ID used to identify different accounts in the same app
- **Data.AppInfo.Name** - Name of the service in use (Profile card)
- **Data.bandwidthEstimateMbps** - Effective bandwidth estimate in Mbps
- **Data.cardCorrelationId** - The globally unique identifier for a persona card
- **Data.cardPersonaCorrelationId** - The globally unique identifier for a specific persona shown in a card
- **Data.clientCorrelationId** - The globally unique identifier for the app's session
- **Data.clientType** - The type of device the app is run on.
- **Data.eventId** - Name identifier of the event, for example "LivePersonaCardRenderedAction"
- **Data.exportName** - Human readable name of the user action event, for example "OpenedPersonaCard"
- **Data.exportType** - Category of the event for GDPR export request

- **Data.externalAppSessionCorrelationId** - A globally unique identifier for the app to identify all persona cards opened in the same sub-session
- **Data.feature** - Used to group various events of the same feature (Profile card)
- **Data.hasPersonaInsightRing** - Insights from Office or LinkedIn could be available for the user
- **Data.hostAppRing** - The ring by which the app was distributed
- **Data.immersiveProfileCorrelationId** - A globally unique identifier for the expanded profile view session
- **Data.OTelJS.Version** - Version of OTel logger
- **Data.personaCorrelationId** - A globally unique identifier for unique personas in a session
- **Data.properties** - Additional metadata collected for each event as follows. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
  - **cardCorrelationId** - Duplicate of Data.appContextId above
  - **cardPersonaCorrelationId** - Duplicate of Data.cardCorrelationId above
  - **consumerCorrelationId** - Duplicate of Data.clientCorrelationId above
  - **networkEffectiveType** - The effective type of network connection, for example "slow-2g Online" to identify whether the user is connected to the internet at the time of showing the persona card
  - **networkType** - The type of network connectivity of the device in use
  - **roundTripEstimateMs** - Estimated effective round trip of the current connection in milliseconds
- **Data.region** -The geographical region of the profile card backend service to which user is connected
- **Data.tenantAadObjectId** - The tenant to which a user's subscription is tied. Allows us to classify issues and identify whether a problem is widespread or isolated to a set of users or a specific tenant
- **Data.type** -Type of the logged event, for example, Trace, Error, Event
- **Data.userAadObjectId** -The globally unique user identifier for an enterprise Microsoft account (duplicate of Data.UserInfo.Id)
- **Data.UserInfo.Id** - The globally unique user identifier for an enterprise Microsoft account
- **Data.UserInfo.MsaId** - The globally unique user identifier for a consumer Microsoft account
- **Data.UserInfo.OMSTenantId** - The tenant that a user's subscription is tied to. Allows us to classify issues and identify whether a problem is widespread or isolated to a set of users or a specific tenant
- **Data.userPuid** -The globally unique user identifier for a consumer Microsoft account (duplicate of Data.UserInfo.MsaId)
- **Data.version** -The version of the service (Profile Card)
- **Data.viewType** -Defines the type of the Profile card displayed
- **Data.wasOpenedAsCompactCard** - Used to identify if the card was opened as a compact view initially
- **NetworkCost** - Indicates network cost/type (metered, metered above cap, etc.)
- **NetworkCountry** - The Country Code of the Sender, based on the unscrubbed Client IP Address.

Office.Manageability.Client Fetch.PolicyPreChecks

Critical telemetry to track failure\success for cloud policy fetch precheck validation. ExitReason contains an enumerator map to the pre-check condition that failed.

The following fields are collected:

- **Data.ExitReason** - An enumerator value telling the exit reason, if the Precheck failed
- **Data.Log** - Custom log message indicating the precheck success or failure

Office.Manageability.Client.Fetch.AndApplyPolicy

Critical telemetry to track failure\success for cloud policy fetch initiation from app. Exit Reason contains an enumerator Map to the failure reason.

The following fields are collected:

- **Data.ExitReason** - An enumerator value telling the exit reason, if the Precheck failed
- **Data.Log** - Custom log message indicating the precheck success or failure

Office.OfficeMobile.Fluid.FluidFileOperations

This event is collected for Office applications when a fluid file operation takes place. Data is used to track feature health and understand user-experience based on the operation information.

The following fields are collected:

- **FailureReason** - If the operation was a failure. Contains the error code of the failure.
- **Result** - A boolean value that indicates the end result of the operation.
- **Type** - The operation type (for example, Open).

Office.OfficeMobile.PdfViewer.PdfFileOperations (on Android)

The event is collected for the Office app for Android. It records when a .pdf open, close, or save operation takes place and is used to understand and prioritize the user experience based on .pdf file operation information. The event enables us to keep the .pdf open, close, and save operations performing as expected, and to improve .pdf file operation performance.

The following fields are collected:

- **Data_Doc_FileOpSessionID** - Unique ID for a Document Session
- **Data_ErrorCode** - error in case of file open failures/download failures / download canceled
- **Data_ErrorMessage** - relevant message-to-error code
- **Data_FailureReason** - In case of open failure, these enums define the reason for failure.
- **Data_FetchReason** - Denotes how the file was fetched (manual, cached, not cached)
- **Data_FileGUID** – Global identifier for the file, which is randomly generated
- **Data_FileLocation** - Location where the file sits, ex: Local, ODSP, iCloud, etc.
- **Data_FileOpenEntryPoint** - entry point for file open
- **Data_FileSize** - Size of the file on which the operation is happening
- **Data_NetworkRequestErrorResponse** – Network Error response corresponding to error code.
- **Data_NetworkRequestStage** – Error stage in case of cloud pdf file download.
- **Data_OpenMode** - In which mode the PDF was opened, ex: 0: View mode, 2: Sign mode
- **Data_PageCount** - Count of page in the PDF File.
- **Data_PasswordProtected** - Marker that tells whether the file is password protected or not.
- **Data_ProviderApp** - currently providing provider app in case of file activation only

- **Data_ReadOnly** - Marker that tells whether the file is read only or not.
- **Data_Result** - The status of the operation being performed, ex: true:success, false:failure
- **Data_Type** - Type of file operation (open, close, or save)

Office.OfficeMobile.PdfViewer.PdfFileOperations (on iOS)

The event is collected for the Office app for iOS. It records when a .pdf open, close, or save operation takes place and is used to understand and prioritize the user experience based on .pdf file operation information. The event enables us to keep the .pdf open, close, and save operations performing as expected, and to improve .pdf file operation performance.

- **Data_Doc_FileOpSessionID** - Unique ID for a Document Session
- **Data_ErrorCode** – error in case of file open failures/download failures / download canceled
- **Data_ErrorMessage** – relevant message to error code
- **Data_FailureReason** - In case of open failure, these enums define the reason for failure.
- **Data_FetchReason** - Denotes how the file was fetched (manual, cached, not cached)
- **Data_FileGUID** – Global identifier for the file, which is randomly generated
- **Data_FileLocation** - Location where the file sits (Local, ODSP, iCloud, etc.)
- **Data_FileOpenEntryPoint** – entry point for file open
- **Data_FileSize** - Size of the file on which the operation is happening
- **Data_OpenMode** - In which mode the PDF was opened (0: View mode 2: Sign mode)
- **Data_PageCount** - Count of page in the PDF File.
- **Data_PasswordProtected** - Marker that tells whether the file is password protected or not.
- **Data_ProviderApp** – currently providing provider app in case of file activation only
- **Data_ReadOnly** - Marker that tells whether the file is read only or not.
- **Data_Result** - The status of the operation being performed (true:success, false:failure)
- **Data_Type** - Type of file operation (open, close, or save)

Office.OfficeMobile.Search.VoiceSearchUsage

This event is triggered when the user taps on the microphone in the search box inside the Office Mobile app. The event will track the usage of voice search and also time taken to establish service request post tap on microphone. This data will be used to track the usage and health of the feature.

The following fields are collected:

- **VoiceButtonClicked** - Integer value mapped to taps on the voice search mic.
- **VoiceConsentAccepted** - Integer value mapped to Cortana consent/permissions given (only applicable to Microsoft internal audience)
- **VoicePermissionGranted** - Integer value mapped to the permissions access action
- **VoiceRecognitionCompleted** - Integer value mapped to successful voice recognition completion
- **VoiceSearchError** - Integer value mapped to occurrence of errors during speech to text.
- **VoiceSearchStartupLatency** - Real number mapped to start-up latency for speech.

- **VoiceSearchTokenFetchingLatency** - Real number mapped to successful token fetch latency

Office.OneNote.Android.App.Navigation.NavigationUIStateChanged

[This event was previously named OneNote.App.Navigation.NavigationUIStateChanged.]

This event collects the critical signal used to ensure OneNote users can successfully navigate through the app. The telemetry is used to ensure critical regression detection for OneNote app and service health.

The following fields are collected:

- **IS_SPANNED** - Indicates whether the app is in a spanned mode. This is specifically logged for foldable devices.
- **NEW_STATE** - Indicates the applications' state right after the navigation
- **OLD_STATE** - Indicates the applications' state right before the navigation

Office.OneNote.Android.Canvas.PageOpened

[This event was previously named OneNote.Canvas.PageOpened.]

The signal used to record when a Page is opened. The telemetry is used to monitor, detect, and fix any issues caused when a Page is opened in OneNote

The following fields are collected:

- **JOT_ID** - object of the page opened
- **TIME_TAKEN_IN_MS** - time taken to open page

Office.OneNote.Android.Capture.NewNote.NewNoteTaken

[This event was previously named OneNote.Capture.NewNote.NewNoteTaken.]

This signal is used to ensure that after a user signs-into a OneNote Android App, notebooks are properly provisioned, and user has successfully created a new note. This is used to ensure critical regression detection for OneNote app and service health.

The following fields are collected:

- None

Office.OneNote.Android.LensSDK.OfficeLensLaunched

[This event was previously named OneNote.LensSDK.OfficeLensLaunched.]

This event collects the critical signal used to ensure that OfficeLens is launched correctly. The telemetry is used to ensure critical regression detection for OneNote app and service health.

The following fields are collected:

- **CAPTURE_MODE** - Indicates which mode has OfficeLens has been launched in. It could be default, edit, fast insert or video import.
- **ERROR_CODE** - Indicates the launch error code in case there was an error while launching.
- **IMAGE_COUNT** - Indicates the number of images taken
- **LAUNCH_REASON** - Indicates the flow under which OfficeLens was launched. It could be over the lock screen or via Camera or Gallery options in StickyNotes or via OneNote Canvas etc.

Office.OneNote.Android.MessageBar.MessageBarClicked

[This event was previously named OneNote.MessageBar.MessageBarClicked.]

The signal used to indicate any issues encountered while using Message Bar. The telemetry is used to monitor, detect, and fix any issues caused during interaction with Message Bar

The following fields are collected:

- **Message_Bar_Type** - Returns if the user is using old or new message bar
- **Message_Type** - Returns the error message ID

Office.OneNote.Android.StickyNotes.NoteCreated

Critical signal that is used to monitor the ability of Sticky Notes users to create notes in the app. Telemetry is used to ensure critical regression detection for OneNote app and service health. If users can't create a note, this would trigger a high severity incident.

The following fields are collected:

- **IsExportable** - A flag indicating whether this event was a result of a user action or not. Should be set to True as NoteCreated is a user-triggered action.
- **NoteLocalId** - Distinguishable unique identifier assigned to a note at the time of a user creates the note within the app.
- **StickyNotes-SDKVersion** - Version number indicating the version of Sticky Notes the user is using. Allows us to identify which versions of the product are showing an issue so that we can correctly prioritize it.

Office.OneNote.Android.StickyNotes.NoteViewed

Critical signal that is used to monitor the ability of Sticky Notes users to view notes in the app. Telemetry is used to ensure critical regression detection for OneNote app and service health. If users can't view their notes, this would trigger a high severity incident.

The following fields are collected:

- **HasImages** - A flag indicating whether the note viewed has images stored in it.
- **IsExportable** - A flag indicating whether this event was a result of a user action or not. Should be set to True as NoteViewed is a user-triggered action.
- **NoteLocalId** - Distinguishable unique identifier assigned to a note at the time a user creates the note within the app.
- **StickyNotes-SDKVersion** - Version number indicating the version of Sticky Notes the user is using. Allows us to identify which versions of the product are showing an issue so that we can correctly prioritize it.

Office.OneNote.Canvas.Ink.InkStrokeLogger

This event is used to detect and diagnose a high frequency bug that a user encounters while using Ink feature. This will be used to determine the most appropriate mode of fixing this issue.

The following fields are collected:

- **CurrentCanvasZoomFactor** - Current Zoom factor of the canvas.
- **CurrentNotebook** - Identifier of the current active notebook.
- **CurrentPage** - Identifier of the current active page
- **CurrentSection** - Identifier of the current active section.
- **DefaultCanvasZoomFactor** - Default Zoom factor of the canvas.
- **InkStrokeCount** - Total count of ink strokes since the last log.
- **InkStrokeWithLayerInkEffect** - Count of ink strokes with layer ink effect since the last log.

- **InkStrokeWithoutPressureCount** - Count of Ink Strokes without Pressure since the last log.
- **InkStrokeWithPencilInkEffect** - Count of ink strokes with pencil ink effect since the last log.
- **InkStrokeWithTilt** - Count of ink strokes with tilt since the last log.

Office.OneNote.Navigation.CreatePage

Critical signal used to monitor the ability of OneNote users to create pages in OneNote. Telemetry used to ensure critical regression detection for OneNote app and service health. If users can't create a page this would trigger a high severity incident.

The following fields are collected:

- **IsAtSectionEnd** - Indicates whether a new page is created at the end of section or not.
- **IsBlank** - Indicates whether a new page is blanked page or created with a template.
- **IsRecentsView** - Indicates whether a page is created from a recents or not.
- **NavView** - Indicates whether a page is created from a navigation view or not.
- **NoteType** - Indicates the type (quick note, list, or photo) of a page.
- **QuickNoteType** - Indicates the type (quick note, list, or photo) of a page.
- **RailState** - Indicates the state of OneNote's navigation rail when creating a page.
- **Trigger** - Indicates an entry point where the create page action is started.
- **TriggerInfo** - Indicates additional information related to the trigger.

Office.OneNote.Navigation.CreateSection

Critical signal used to monitor the ability of OneNote users to create sections in OneNote. Telemetry used to ensure critical regression detection for OneNote app and service health. If users can't create a page this would trigger a high severity incident.

The following fields are collected

- **NotebookID** - A unique identifier of a notebook.
- **SectionID** - A unique identifier of a section created.
- **Trigger** - Indicates an entry point where the create section action is started.
- **TriggerInfo** - Indicates additional information related to the trigger.

Office.OneNote.Navigation.Navigate

Critical signal used to monitor the ability of OneNote users to navigate between pages in OneNote. Telemetry used to ensure critical regression detection for OneNote app and service health. If users can't navigate this would trigger a high severity incident.

The following fields are collected:

- **FromNotebook** - A unique identifier of a notebook.
- **FromPage** - A unique identifier of a page.
- **FromSection** - A unique identifier of a section.
- **FromSectionGroup** - A unique identifier of a section group.
- **IsCurrentUserEduStudent** - Indicates whether the current user has a student role in an education notebook or not.

- **IsEduNotebook** - Indicates whether the current page is in an education notebook or not.
- **IsEduNotebookReadOnlyPage** - Indicates whether the current page is a read only page in an education notebook or not.
- **ToNotebook** - A unique identifier of a notebook.
- **ToPage** - A unique identifier of a page.
- **ToSection** - A unique identifier of a section.
- **ToSectionGroup** - A unique identifier of a section group.

Office.OneNote.NotebookManagement.CreateNotebook

Critical signal used to monitor the ability of OneNote users to create notebooks in OneNote. Telemetry used to ensure critical regression detection for OneNote app and service health. If users can't create notebooks this would trigger a high severity incident.

The following fields are collected:

- **NotebookID** - A unique identifier of a notebook.

Office.OneNote.NotebookManagement.OpenNotebook

Critical signal used to monitor the ability of OneNote users to open notebooks in OneNote. Telemetry used to ensure critical regression detection for OneNote app and service health. If users can't open notebooks this would trigger a high severity incident.

The following fields are collected:

- **NotebookID** - A unique identifier of a notebook.

Office.OneNote.Search.Search

Critical signal ID used to monitor the ability of OneNote users to find information across thousands of pages and notebooks. Telemetry used to ensure critical regression detection for OneNote app and service health. If users can't find information across notebooks this would trigger a high severity incident.

The following fields are collected:

- **PageSearchResultCount** - Indicates the number of search's results found in a page search mode.
- **PageTimeToFirstResultInMs** - Indicates the amount of time OneNote takes to find the first match in a page search mode.
- **PageTimeToLastResultInMs** - Indicates the amount of time OneNote takes to find the last match in a page search mode.
- **PageTimeToMedianResultInMs** - Indicates the median of time OneNote takes to find all matches in a page search mode.
- **SearchResultCount** - Indicates the number of search's results found.
- **TagSearchResultCount** - Indicates the number of search's results found in a tag search mode.
- **TagTimeToFirstResultInMs** - Indicates the amount of time OneNote takes to find the first match in a tag search mode.
- **TagTimeToLastResultInMs** - Indicates the amount of time OneNote takes to find the last match in a tag search mode.
- **TagTimeToMedianResultInMs** - Indicates the median of time OneNote takes to find all matches in a tag search mode.

- **TimeToFirstResultInMs** - Indicates the amount of time OneNote takes to find the first match.
- **TimeToLastResultInMs** - Indicates the amount of time OneNote takes to find the last match.
- **TimeToMedianResultInMs** - Indicates the median of time OneNote takes to find all matches.

Office.OneNote.SIGS.CriticalErrorEncountered

This event captures a critical signal that is used to monitor the health of Signal Ingestion Service (SIGS), by logging whenever a critical error is encountered. Critical errors can block the whole of SIGS, and this will help us catch any such issues as soon as they are encountered by users.

Without this, we will be dependent on users to report the problems they are facing. Absence of such telemetry would make the turnaround time for such issues much higher.

The following fields are collected:

- **ErrorCode** - The code of the issue that was hit by the user.

Office.OneNote.StickyNotes.NoteCreated (on iOS), OneNote.StickyNotes.NoteCreated (on Android)

This is a critical signal that is used to monitor the ability of Sticky Notes users to create notes in the app. Telemetry is used to ensure critical regression detection for OneNote app and service health. If users can't create a note, this would trigger a high severity incident.

The following fields are collected:

- **NoteLocalId** - Distinguishable unique identifier assigned to a note at the time of a user creates the note within the app.
- **IsExportable** - A flag indicating whether this event was a result of a user action or not. Should be set to True as NoteCreated is a user-triggered action.
- **StickyNotes-SDKVersion** - Version number indicating the version of Sticky Notes the user is using. Allows us to identify which versions of the product are showing an issue so that we can correctly prioritize it.

Office.OneNote.StickyNotes.NoteViewed (on iOS), OneNote.StickyNotes.NoteViewed (on Android)

This is a critical signal that is used to monitor the ability of Sticky Notes users to create notes in the app. Telemetry is used to ensure critical regression detection for OneNote app and service health. If users can't create a note, this would trigger a high severity incident.

The following fields are collected:

- **HasImages** - A flag indicating whether the note viewed has images stored in it.
- **IsExportable** - A flag indicating whether this event was a result of a user action or not. Should be set to True as NoteViewed is a user-triggered action.
- **NoteLocalId** - Distinguishable unique identifier assigned to a note at the time a user creates the note within the app.
- **StickyNotes-SDKVersion** - Version number indicating the version of Sticky Notes the user is using. Allows us to identify which versions of the product are showing an issue so that we can correctly prioritize it.

Office.OneNote.Storage.NotebookSyncResult

This event logs notebook sync result. It is used for figuring out how many unique sync targets when calculating OneNote sync score.

The following fields are collected

- **CachedError_Code** - a numbered or alphanumeric code used to determine the nature of the cached

error, and/or why it occurred

- **CachedError_Description** -a description of the cached error
- **CachedError_Tag** -indicate where in the code throws the cached error
- **CachedError_Type** -the type of the cached error, for example, Win32Error, etc.
- **ExecutionTime** -time in milliseconds taken to replicate the notebook
- **Gosid** -global object space ID
- **IdentityType** -identity type, for example, Windows Live, Org ID, etc.
- **InitialReplicationInSession** -is this replication the first notebook replication after open or not
- **IsBackgroundSync** -is this a background sync or not
- **IsCachedErrorSuppressed** -is the cached error suppressed or not
- **IsCachedErrorUnexpected** -is the cached error unexpected or not
- **IsNotebookErrorSuppressed** -is the notebook level sync error suppressed or not
- **IsNotebookErrorUnexpected** -is the notebook level sync error unexpected or not
- **IsSectionErrorSuppressed** -is the section sync error suppressed or not
- **IsSectionErrorUnexpected** -is the section sync error unexpected or not
- **IsUsingRealtimeSync** -is the notebook sync using modern page content sync or not
- **LastAttemptedSync** -timestamp when the notebook was attempted to be synced last time
- **LastBackgroundSync** -timestamp when the latest background sync was attempted
- **LastNotebookViewedDate** -the date when the notebook was last viewed
- **LastSuccessfulSync** -timestamp when the notebook successfully synced before
- **NeedToRestartBecauseOfInconsistencies** -does the sync need to restart because of inconsistencies or not
- **NotebookErrorCode** -notebook level sync error code saved on notebook graph space
- **NotebookId** -notebook ID
- **NotebookType** -notebook type
- **ReplicatingAgainBecauseOfInconsistencies** -does the sync restart because of inconsistencies or not
- **SectionError_Code** -a numbered or alphanumeric code used to determine the nature of the section sync error, and/or why it occurred
- **SectionError_Description** -a description of the section sync error
- **SectionError_Tag** -indicate where in the code throws the section sync error
- **SectionError_Type** -the type of the section sync error, for example, Win32Error, etc.
- **Success** -is the notebook sync successful or not
- **SyncDestinationType** -sync destination type, that is, OneDrive or SharePoint Online
- **SyncId** -a number unique to each notebook sync

- **SyncWasFirstInSession** -is this sync the first sync in current session
- **SyncWasUserInitiated** -is this sync user initiated or not
- **TenantId** -SharePoint tenant ID
- **TimeSinceLastAttemptedSync** -time since last notebook sync attempt
- **TimeSinceLastSuccessfulSync** -time since last successful notebook sync

Office.OneNote.System.AppLifeCycle.AppLaunch

The critical signal used to ensure OneNote users can successfully launch the app. The telemetry is used to ensure critical regression detection for OneNote app and service health. If users can't launch the app in our performance window, this would trigger a high severity incident.

The following fields are collected: None

Office.Outlook.Desktop.AccountConfiguration.CreateAccountResult

Result of adding an account to Outlook in a new profile, from the Office Backstage, or from the account settings dialog. The data is actively monitored to ensure we don't see any spikes in failures. We also analyze the data to find areas of improvement. We aim to improve this success rate with each release.

The following fields are collected:

- **AccountCreationResult** – The result (success, failure, cancellation, etc.) of adding the account to Outlook.
- **AccountCreationTime** – time taken to attempt account creation
- **AccountInfoSource** - account settings source (for example, AutoDiscover, GuessSmart, AutoDetect, etc.)
- **AccountType** – The type of account being configured.
- **HashedEmailAddress** – hashed email address
- **ShowPasswordPageFlightEnabled** - indicator if ShowPopImapPasswordPage flight is enabled

Office.Outlook.Desktop.AccountConfiguration.RepairAccountResult

Result of repairing an account or changing advanced account settings. The data is actively monitored to ensure we don't see any spikes in failures. We also analyze the data to find areas of improvement. Since this a new (refactored) experience we want to make sure we got this right.

The following fields are collected:

- **AccountInfoSource** - account info source for the account used to attempt repair
- **AccountType** - type of account for which account repair was attempted
- **HashedEmailAddress** - hashed email address
- **ManualRepairRequested** - indicator if manual repair was requested
- **Result** - result of attempt to repair account. For example: "Success" or "Fail_SaveChangesToAccount"

Office.Outlook.Desktop.AccountConfiguration.UpdatePasswordResult

Result of updating an account's password from the Account Settings dropdown. The data is actively monitored to ensure we don't see any spikes in failures. We also analyze the data to find areas of improvement. Since this a new (refactored) experience we want to make sure we got this right.

The following fields are collected:

- **AccountType** - type of account for which updating password was attempted

- **HashedEmailAddress** - hashed email address
- **Result** - result of attempt to update password. For example: "Success" or "Fail_AllowLessSecureAppsDisabled"

Office.Outlook.Desktop.Stores.CreateNewStore

Collects the result of creating a new store (with type and version), and the result code. We actively monitor this event to track the health a user's ability to sync and store mail locally, archive mails (in a PST), or use Groups.

The following fields are collected:

- **Standard HVA Activity** with custom payload
- **StoreType** – The type of store created OST/PST/NST
- **StoreVersion** – The store version created Small/Large/Tardis

Office.Outlook.Mac.AccountAddWorkflow

Result of adding an account in Outlook. The data is monitored to ensure we don't see any spikes in failures. We also analyze the data to find areas of improvement. We aim to improve this success rate with each release.

The following fields are collected:

- **AccountConfigMethod** - the account configuration method
- **AccountType** - the type of account being configured
- **AccountWorkflowSession** - session where account workflow is attempted
- **SessionDuration** - duration of session
- **ThreadId** - identifier for the thread

Office.Outlook.Mac.AccountOnboardingFlow

Result of adding an account in Outlook using new account configuration experience. The data is monitored to ensure we don't see any spikes in failures. We also analyze the data to find areas of improvement. We aim to improve this success rate with each release.

The following fields are collected:

- **AccountConfigAutoSignIn** - automatic account configuration set by admin
- **AccountConfigDomain** - domain specified during account configuration
- **AccountConfigEntryPoint** - entry point where user entered account configuration
- **AccountConfigErrorCode** - error code encountered during account configuration
- **AccountConfigErrorString** - error encountered during account configuration
- **AccountConfigMethod** - account configuration method
- **AccountConfigPhase** - current step of account configuration workflow
- **AccountConfigPhaseFrom** - beginning step of account configuration workflow
- **AccountConfigPhaseTo** - last step of account configuration workflow
- **AccountType** - type of account being configured
- **AccountWorkflowSession** - session where account workflow is attempted
- **SessionDuration** - duration of session

Office.Outlook.Mac.DeleteAccountUsage

Result of deleting an account in Outlook. The data is monitored to ensure we don't see any spikes in failures. We also analyze the data to find areas of improvement. We aim to improve this success rate with each release.

The following fields are collected:

- **AccountType** - type of account being configured
- **AccountID** - account identifier
- **DeprovisionAccount** - indicates whether account is removed from server
- **IsFastDelete** - indicates whether account is deleted on background thread

Office.PowerPoint.DocOperation.Close

Collected when PowerPoint presentations are closed. It contains the information needed to be able to properly investigate and diagnose issues that happen through the close process which entail persisting and syncing the user's data. Microsoft uses this data to ensure that close is working as expected and user content is successfully being persisted.

The following fields are collected:

- **Data_AddDocTelemetryResult:long** - Does this log entry have all necessary document telemetry (Data_Doc_* fields)? If not, why?
- **Data_AutoSaveDisabledReasons:string** - Predefined set of values of why was autosave disabled on this document? (Merge error, Save error, Group policy etc.)
- **Data_CloseReason:long** - How was close performed? Closing document? Closing app?
- **Data_CppUncaughtExceptionCount:long** - Number of unhandled exceptions
- **Data_DetachedDuration:long** - Time for which Activity was detached/not running
- **Data_Doc_AccessMode:long** - How was this document opened (Read only | read write)
- **Data_Doc_AssistedReadingReasons:long** - Predefined set of values of why document was opened in assisted reading mode
- **Data_Doc_AsyncOpenKind:long** – Indicates whether a cached version of the cloud document was opened and which asynchronous refresh logic was used.
- **Data_Doc_ChunkingType:long** - How is document stored in SharePoint
- **Data_Doc_EdpState:long** - Enterprise Data Protection state of document
- **Data_Doc_Ext:string** - Document extension
- **Data_Doc_Extension:string** - Document extension
- **Data_Doc_FileFormat:long** - Predefined set of values of format of file (more granular than extension)
- **Data_Doc_Fqdn:string** - Where is document stored (SharePoint.com, live.net), only available for Office 365 domains
- **Data_Doc_FqdnHash:string** - Hash of where document is stored
- **Data_Doc_IdentityTelemetryId:string** - Unique GUID of user
- **Data_Doc_IdentityUniqueId:string** - Unique identifier of identity that was used for Shared Documents action
- **Data_Doc_IOFlags:long** - Bitmask for various IO-related flags for a given document

- **Data_Doc_IrmRights:long** - Predefined set of values of what type of Information Rights Management is applied on this document (Forward, Reply, SecureReader, Edit etc.)
- **Data_Doc_IsCloudCollabEnabled:bool** - True if the "IsCloudCollabEnabled" HTTP header has already been received from an OPTIONS request.
- **Data_Doc_IsIncrementalOpen:bool** - Was document opened incrementally (new feature that opens document without needing to download entire document)
- **Data_Doc_IsOcsSupported:bool** - Is Document supports co-authoring using new OCS service
- **Data_Doc_IsOpeningOfflineCopy:bool** - verifies if document is being opened from local cache
- **Data_Doc_IsSyncBacked:bool** - verifies if document is being opened from folder that is using OneDrive sync back app
- **Data_Doc_Location:long** - Predefined set of values of where document is stored (Local, SharePoint, WOPI, Network etc.)
- **Data_Doc_LocationDetails:long** - Predefined set of values of more detailed location (Temp folder, downloads folder, One Drive Documents, One Drive Pictures etc.)
- **Data_Doc_NumberCoAuthors:long** - Number of co-authors at the time of opening of a document
- **Data_Doc_PasswordFlags:long** - Predefined set of values of how document is encrypted with password (None, password to read, password to edit)-
- **Data_Doc_ReadOnlyReasons:long** - Predefined set of values of why this document was marked read only (Locked on server, final document, password protected to edit etc.)
- **Data_Doc_ResourceIdHash:string** - Hash of resource identifier for documents stored in cloud
- **Data_Doc_RtcType** - Indicates how the real-time channel (RTC) was set up for current file (Disabled, unsupported, on demand, always on, etc.).
- **Data_Doc_ServerDocId:string** - immutable identifier for documents stored in cloud
- **Data_Doc_ServerProtocol:long** - Predefined set of values of which protocol is used to talk to server (Http, Cobalt, WOPI etc.)
- **Data_Doc_ServerType:long** - Predefined set of values of type of server (SharePoint, DropBox, WOPI)
- **Data_Doc_ServerVersion:long** - verifies if server is based off Office14, Office15, or Office 16
- **Data_Doc_SessionId:long** - generated GUID that Identifies the instance of the document within the same process session
- **Data_Doc_SharePointServiceContext:string** - An opaque string, typically GridManagerID.FarmID. Useful for correlating client-side and server-side log
- **Data_Doc_SizeInBytes:long** - Document size in bytes
- **Data_Doc_SpecialChars:long** - Bitmask indicating special chars in the document's URL or Path
- **Data_Doc_StorageProviderId:string** - A string that identifies the document's storage provider, like "DropBox"
- **Data_Doc_StreamAvailability:long** - Predefined set of values of status of document Stream(available, permanently disabled, not available)
- **Data_Doc_UrlHash:string** - hash of full URL of documents stored in cloud

- **Data_Doc_UsedWrsDataOnOpen:bool** - true if the file was opened incrementally using pre cached WRS data on the host
- **Data_Doc_WopiServiceId:string** - WOPI Service identifier, for example "Dropbox"
- **Data_DocHasStorage:bool** - Does this document have local storage?
- **Data_fLifeguarded:bool** - Was document ever lifeguarded (feature to fix document errors by themselves without prompting user)?
- **Data_IsDocAutoSaveable:bool** - Is presentation auto saveable?
- **Data_IsDocDirty:bool** - Does presentation have changes that are not yet saved?
- **Data_IsNewDoc:bool** - Is new document or existing
- **Data_IsRecoveredDoc:bool** - Is document recovered one? (If prior session crashed, we show document recovery pane on next session)
- **Data_NewDocDiscarded:bool** - Was new presentation discarded without being saved
- **Data_OCSClosingDlgCanceled:bool** - If upload is pending on OCS while user closes document, dialog is popped up to user to wait. Which option user chose?
- **Data_OCSClosingDlgExpired:bool** - Did dialog expire (after 1 minute) on its own?
- **Data_OCSClosingStatus:long** - What's final status of OCS (In CSI, Closable, In OCS Transition, In CSI transition, etc.)
- **Data_OCSClosingWaitDurationMS:long** - How much time user had to wait for OCS to upload
- **Data_OCSHandleTransitionResult:long** - Predefined set of values of result of transition performed during close (Already tried, continue to close, etc.)
- **Data_ServerDocId:string** - GUID to uniquely identify a document
- **Data_StopwatchDuration:long** - Total time for Activity
- **Data_UserContinuedZRTClose:bool** - Upon showing dialog on close, did user selected 'Continue' to close?

Office.PowerPoint.DocOperation.NewDocument

Collected when PowerPoint creates a new presentation. Includes success failure and performance metrics.

This information is used to ensure we can create files successfully and with no degradation in performance.

The following fields are collected:

- **NewDocumentType** – Whether new document is created from template or just created blank?
- **FLifeguarded** – Is document life guarded (feature that restores corrupt document state without prompting user)

Office.PowerPoint.DocOperation.OpenCompleteProtocol

Collected when PowerPoint opens presentations. It contains the information needed to be able to properly investigate and diagnose issues that happen through the end stages of the open process.

Microsoft uses this data to ensure the feature is working as expected and there is no degradation to opening presentations.

The following fields are collected:

- **Data_AntiVirusScanMethod:long** - Predefined set of values of type of AntiVirus scanned (IOAV, AMSI,

None etc.)

- **Data_AntiVirusScanStatus:long** - Predefined set of values of anti-virus scan that happens for every document opened (NoThreatsDetected, Failed, MalwareDetected etc.)
- **Data_CloseAndReopen:bool** - Was this document closed and reopened?
- **Data_ClpDocHasDrmDoc:bool** - Whether the document has a DRM document
- **Data_ClpDocHasIdentity:bool** - Whether the document has identity info (used to get and set sensitivity labels)
- **Data_ClpDocHasSessionMetadata:bool** – Whether the document has working sensitivity label metadata from the session
- **Data_ClpDocHasSpoMetadata:bool** - Whether the document has sensitivity label metadata from SPO via IMetadataCache
- **Data_ClpDocHasSpoPackage:bool** – Whether the document has sensitivity label metadata from SPO via IPackage
- **Data_ClpDocIsProtected:bool** - Whether or not the document is protected by IRM
- **Data_ClpDocMetadataSource:int** - Enum specifying where sensitivity label metadata is from (IRM, OPC part, Sharepoint etc)
- **Data_ClpDocNeedsUpconversion:bool** - Whether the document needs to upconvert sensitivity label data from the custom.xml part
- **Data_ClpDocNumFailedSetLabels:int** - Count of sensitivity labels that failed to set on the document
- **Data_ClpDocSessionMetadataDirty:bool** - Whether the document has working sensitivity label metadata that has been dirtied
- **Data_ClpDocWasInTrustBoundary:bool** – Whether the document was in the trust boundary (allowing for coauthoring on documents protected by sensitivity labels)
- **Data_DetachedDuration:long** - Time for which Activity was detached/not running
- **Data_Doc_AccessMode:long** - How was this document opened (Read only | read write)
- **Data_Doc_AssistedReadingReasons:long** - Predefined set of values of why document was opened in assisted reading mode
- **Data_Doc_AsyncOpenKind:long** – Indicates whether a cached version of the cloud document was opened and which asynchronous refresh logic was used.
- **Data_Doc_ChunkingType:long** - How is document stored in SharePoint
- **Data_Doc_EdpState:long** - Enterprise Data Protection state of document
- **Data_Doc_Ext:string** - Document extension
- **Data_Doc_Extension:string** - Document extension
- **Data_Doc_FileFormat:long** - Predefined set of values of format of file (more granular than extension)
- **Data_Doc_Fqdn:string** - Where is document stored (SharePoint.com, live.net), only available for Office 365 domains
- **Data_Doc_FqdnHash:string** - Hash of where document is stored
- **Data_Doc_IdentityTelemetryId:string** - Unique GUID of user

- **Data_Doc_IdentityUniqueId:string** - Unique identifier of identity that was used for Shared Documents action
- **Data_Doc_IOFlags:long** - Bitmask for various IO-related flags for a given document
- **Data_Doc_IrmRights:long** - Predefined set of values of what type of Information Rights Management is applied on this document (Forward, Reply, SecureReader, Edit etc.)
- **Data_Doc_IsCloudCollabEnabled:bool** - True if the "IsCloudCollabEnabled" HTTP header has already been received from an OPTIONS request.
- **Data_Doc_IsIncrementalOpen:bool** - Was document opened incrementally (new feature that opens document without needing to download entire document)
- **Data_Doc_IsOcsSupported:bool** - Is Document supports co-authoring using new OCS service
- **Data_Doc_IsOpeningOfflineCopy:bool** - Is document being opened from local cache?
- **Data_Doc_IsSyncBacked:bool** - Is document opened from folder that is using OneDrive sync back app
- **Data_Doc_Location:long** - Predefined set of values of where document is stored (Local, SharePoint, WOPI, Network etc.)
- **Data_Doc_LocationDetails:long** - Predefined set of values of more detailed location (Temp folder, downloads folder, One Drive Documents, One Drive Pictures etc.)
- **Data_Doc_NumberCoAuthors:long** - Number of co-authors at the time of opening of a document
- **Data_Doc_PasswordFlags:long** - Predefined set of values of how document is encrypted with password (None, password to read, password to edit)-
- **Data_Doc_ReadOnlyReasons:long** - Predefined set of values of why this document was marked read only (Locked on server, final document, password protected to edit etc.)
- **Data_Doc_ResourceIdHash:string** - Hash of resource identifier for documents stored in cloud
- **Data_Doc_RtcType** - Indicates how the real-time channel (RTC) was set up for current file (Disabled, unsupported, on demand, always on, etc.).
- **Data_Doc_ServerDocId:string** - immutable identifier for documents stored in cloud
- **Data_Doc_ServerProtocol:long** - Predefined set of values of which protocol is used to talk to server (Http, Cobalt, WOPI etc.)
- **Data_Doc_ServerType:long** - Predefined set of values of type of server (SharePoint, DropBox, WOPI)
- **Data_Doc_ServerVersion:long** - verifies if server is based off Office14, Office15, or Office 16
- **Data_Doc_SessionId:long** - generated GUID that Identifies the instance of the document within the same process session
- **Data_Doc_SharePointServiceContext:string** - An opaque string, typically GridManagerID.FarmID. Useful for correlating client side and server-side logs
- **Data_Doc_SizeInBytes:long** - Document size in bytes
- **Data_Doc_SpecialChars:long** - Bitmask indicating special chars in the document's URL or Path
- **Data_Doc_StorageProviderId:string** - A string that identifies the document's storage provider, like "DropBox"
- **Data_Doc_StreamAvailability:long** - Predefined set of values of status of document Stream(available,

permanently disabled, not available)

- **Data_Doc_UrlHash:string** - hash of full URL of documents stored in cloud
- **Data_Doc_UsedWrsDataOnOpen:bool** - true if the file was opened incrementally using pre cached WRS data on the host
- **Data_Doc_WopiServiceId:string** - WOPI Service identifier, for example "Dropbox"
- **Data_ExecutionCount:long** - How many times we executed IncOpen protocol before executing this (OpenComplete) protocol
- **Data_FailureComponent:long** - Predefined set of values of which component caused this protocol to fail? (Conflict, CSI, Internal etc.)
- **Data_FailureReason:long** - Predefined set of values of what's the failure reason (FileIsCorrupt, BlockedByAntivirus etc.)
- **Data_FullDownloadRoundTripCount:long** - The number of roundtrips to the server taken to download the entire document.
- **Data_IsProtocolRunInIncOpenMode:bool** - Was the protocol run for an incremental download, which is a download where parts of the document were downloaded after initially showing it to the user.
- **Data_MethodId:long** - Internally which line of code was last one to be executed
- **Data_StopwatchDuration:long** - Total time for Activity
- **Data_TimeToEdit:long** - Time it took for document to become editable
- **Data_TimeToView:long** - Time it took for first slide of document to be rendered
- **Data_UnhandledException:bool** - Any unhandled native exception?

Office.PowerPoint.DocOperation.Save

Collected whenever PowerPoint performs a save using the modern code path. Includes success or failure result type of save performance metrics and relevant document metadata. Failures in save can result in data loss. Microsoft uses this data to ensure the feature is working as expected and user content is successfully being persisted.

The following fields are collected:

- **Data_AddDocTelemetryResult:long** - Does this log entry have all necessary document telemetry (Data_Doc_* fields)? If not, why?
- **Data_BeforeSaveEvent:long** - Time taken to raise Document Before Save Event
- **Data_CheckDownRevSaveTimeMS:long** - Time taken to check revision
- **Data_CheckMacroSaveTimeMS:long** - Time taken to save macros
- **Data_ClearAutoSaveTimeMS:long** - Time taken to clear AutoSave flag
- **Data_ClearDirtyFlagTimeMS:long** - Time taken to clear document dirty flag
- **Data_CloneDocumentTimeMS:long** - Time taken to clone document before starting the save
- **Data_ClpDocHasDrmDoc:bool** - Whether the document has a DRM document
- **Data_ClpDocHasIdentity:bool** - Whether the document has identity info (used to get and set sensitivity labels)
- **Data_ClpDocHasSessionMetadata:bool** – Whether the document has working sensitivity label

metadata from the session

- **Data_ClpDocHasSpoMetadata:bool** - Whether the document has sensitivity label metadata from SPO via IMetadataCache
- **Data_ClpDocHasSpoPackage:bool** – Whether the document has sensitivity label metadata from SPO via IPackage
- **Data_ClpDocIsProtected:bool** - Whether or not the document is protected by IRM
- **Data_ClpDocMetadataSource:int** - Enum specifying where sensitivity label metadata is from (IRM, OPC part, Sharepoint etc)
- **Data_ClpDocNeedsUpconversion:bool** - Whether the document needs to upconvert sensitivity label data from the custom.xml part
- **Data_ClpDocNumFailedSetLabels:int** - Count of sensitivity labels that failed to set on the document
- **Data_ClpDocSessionMetadataDirty:bool** - Whether the document has working sensitivity label metadata that has been dirtied
- **Data_ClpDocWasInTrustBoundary:bool** – Whether the document was in the trust boundary (allowing for coauthoring on documents protected by sensitivity labels)
- **Data_CommitTransactionTimeMS:long** - Time taken to commit the save transaction
- **Data_CppUncaughtExceptionCount:long** - Uncaught native exceptions while activity was running
- **Data_DetachedDuration:long** - Time for which Activity was detached/not running
- **Data_Doc_AccessMode:long** - How was this document opened (Read only | read write)
- **Data_Doc_AssistedReadingReasons:long** - Predefined set of values of why document was opened in assisted reading mode
- **Data_Doc_AsyncOpenKind:long** – Indicates whether a cached version of the cloud document was opened and which asynchronous refresh logic was used.
- **Data_Doc_ChunkingType:long** - How is document stored in SharePoint
- **Data_Doc_EdpState:long** - Enterprise Data Protection state of document
- **Data_Doc_Ext:string** - Document extension
- **Data_Doc_Extension:string** - Document extension
- **Data_Doc_FileFormat:long** - Predefined set of values of format of file (more granular than extension)
- **Data_Doc_Fqdn:string** - Where is document stored (SharePoint.com, live.net), only available for Office 365 domains
- **Data_Doc_FqdnHash:string** - Hash of where document is stored
- **Data_Doc_IdentityTelemetryId:string** - Unique GUID of user
- **Data_Doc_IdentityUniqueId:string** - Unique identifier of identity that was used for Shared Documents action
- **Data_Doc_IOFlags:long** - Bitmask for various IO-related flags for a given document
- **Data_Doc_IrmRights:long** - Predefined set of values of what type of Information Rights Management is applied on this document (Forward, Reply, SecureReader, Edit etc.)

- **Data_Doc_IsCloudCollabEnabled:bool** - True if the "IsCloudCollabEnabled" HTTP header has already been received from an OPTIONS request.
- **Data_Doc_IsIncrementalOpen:bool** - Was document opened incrementally (new feature that opens document without needing to download entire document)
- **Data_Doc_IsOcsSupported:bool** - Is Document supports co-authoring using new OCS service
- **Data_Doc_IsOpeningOfflineCopy:bool** - verifies if document being is opened from local cache
- **Data_Doc_IsSyncBacked:bool** - Is document opened from folder that is using OneDrive sync back app
- **Data_Doc_Location:long** - Predefined set of values of where document is stored (Local, SharePoint, WOPI, Network etc.)
- **Data_Doc_LocationDetails:long** - Predefined set of values of more detailed location (Temp folder, downloads folder, One Drive Documents, One Drive Pictures etc.)
- **Data_Doc_NumberCoAuthors:long** - Number of co-authors at the time of opening of a document
- **Data_Doc_PasswordFlags:long** -Predefined set of values of how document is encrypted with password (None, password to read, password to edit)
- **Data_Doc_ReadOnlyReasons:long** - Predefined set of values of why this document was marked read only (Locked on server, final document, password protected to edit etc.)
- **Data_Doc_ResourceIdHash:string** - Hash of resource identifier for documents stored in cloud
- **Data_Doc_RtcType** - Indicates how the real-time channel (RTC) was set up for current file (Disabled, unsupported, on demand, always on, etc.).
- **Data_Doc_ServerDocId:string** - immutable identifier for documents stored in cloud
- **Data_Doc_ServerProtocol:long** - Predefined set of values of which protocol is used to talk to server (Http, Cobalt, WOPI etc.)
- **Data_Doc_ServerType:long** - Predefined set of values of type of server (SharePoint, DropBox, WOPI)
- **Data_Doc_ServerVersion:long** - verifies if server is based off Office14, Office15, or Office 16
- **Data_Doc_SessionId:long** - generated GUID that Identifies the instance of the document within the same process session
- **Data_Doc_SharePointServiceContext:string** - An opaque string, typically GridManagerID.FarmID. Useful for correlating client-side and server-side logs
- **Data_Doc_SizeInBytes:long** - Document size in bytes
- **Data_Doc_SpecialChars:long** - Bitmask indicating special chars in the document's URL or Path
- **Data_Doc_StorageProviderId:string** - A string that identifies the document's storage provider, like "DropBox"
- **Data_Doc_StreamAvailability:long** - Predefined set of values of status of document Stream(available, permanently disabled, not available)
- **Data_Doc_UrlHash:string** - hash of full URL of documents stored in cloud
- **Data_Doc_UsedWrsDataOnOpen:bool** - true if the file was opened incrementally using pre cached WRS data on the host
- **Data_Doc_WopiServiceId:string** - WOPI Service identifier, for example "Dropbox"

- **Data_DurationUAEOnSaveStartedMs:long** - Time taken for Unknown App Exit during save
- **Data_EnsureSaveTransactionTimeMS:long** - Time taken to ensure save transaction is created if doesn't exist already
- **Data_FailureComponent:long**- Predefined set of values of which component caused this protocol to fail? (Conflict, CSI, Internal etc.)
- **Data_FailureReason:long** - Predefined set of values of what's the failure reason (FileIsCorrupt, BlockedByAntivirus etc.)
- **Data_fLifeguarded:bool** - Was document ever lifeguarded (feature to fix document errors by themselves without prompting user)?
- **Data_HandleEnsureContentType:long** - Time taken to ensure all the content types are correct
- **Data_HandleEnsureContentTypeTimeMS:long** - Time taken to ensure all the content types are correct
- **Data_HasEmbeddedFont:bool** - Does this document have embedded fonts?
- **Data_InitializeSaveTimeMS:long** - Time taken to initialize document content to begin save
- **Data_InOCSTransition:bool** - Is this save performed for transitioning to OCS
- **Data_IsSavingWithEmbeddedFont:bool** - Does this document have embedded fonts?
- **Data_MethodId:long** - Internally which line of code was last one to be executed
- **Data_PerformEmbedFontsTimeMS:long** - Time taken to serialize embedded fonts
- **Data_PerformModernSaveTimeMS:long** - Time taken to perform modern save (new code)
- **Data_PerformPostSaveTimeMS:long** - Time taken to perform post save functions (notifications, undo entries)
- **Data_PrepareForSaveTimeMS:long** - Time taken to start save
- **Data_RaiseDocumentBeforeSaveEventTimeMS:long** - Time taken to raise the BeforeSave event
- **Data_ReflectDocumentChangeTimeMS:long** - Time taken to reflect saved changes to UI (repopulate thumbnails etc.)
- **Data_ReportStartTimeMS:long** - Time taken to finish initializing telemetry for save
- **Data_ReportSuccessTimeMS:long** - Time taken to finish reporting successful save
- **Data_ResetDirtyFlagOnErrorTimeMS:long** - Time taken to reset document dirty flag on error
- **Data_SaveReason:long** - Predefined set of values of why this save was performed? (AutoSave, ToOCSTransitionSave, ToCSITransitionSave etc.)
- **Data_SaveType:long** - Predefined set of values of save type (SaveAs, Publish, Manual, OMSave etc.)
- **Data_SavingWithFont:bool**- Are we saving document with new embedded fonts?
- **Data_ScrubClonedDocumentTimeMS:long** - Time taken to remove personal information on cloned copy of document
- **Data_StopwatchDuration:long** - Total time for Activity
- **Data_TransactionType:long** - Is it Save or MergeAndSave transaction?

Office.PowerPoint.DocOperation.SaveAs

Collected whenever PowerPoint performs a Save As. Includes success or failure result type of save performance metrics and relevant document metadata. Failures in save can result in data loss. Microsoft uses this data to ensure the feature is working as expected and user content is successfully being persisted.

The following fields are collected:

- **Data_AddDocTelemetryResult:long** - Does this log entry have all necessary document telemetry (Data_Doc_* fields)? If not, why?
- **Data_ClpDocHasDrmDoc:bool** - Whether the document has a DRM document
- **Data_ClpDocHasIdentity:bool** - Whether the document has identity info (used to get and set sensitivity labels)
- **Data_ClpDocHasSessionMetadata:bool** – Whether the document has working sensitivity label metadata from the session
- **Data_ClpDocHasSpoMetadata:bool** - Whether the document has sensitivity label metadata from SPO via IMetadataCache
- **Data_ClpDocHasSpoPackage:bool** – Whether the document has sensitivity label metadata from SPO via IPackage
- **Data_ClpDocIsProtected:bool** - Whether or not the document is protected by IRM
- **Data_ClpDocMetadataSource:int** - Enum specifying where sensitivity label metadata is from (IRM, OPC part, Sharepoint etc)
- **Data_ClpDocNeedsUpconversion:bool** - Whether the document needs to upconvert sensitivity label data from the custom.xml part
- **Data_ClpDocNumFailedSetLabels:int** - Count of sensitivity labels that failed to set on the document
- **Data_ClpDocSessionMetadataDirty:bool** - Whether the document has working sensitivity label metadata that has been dirtied
- **Data_ClpDocWasInTrustBoundary:bool** – Whether the document was in the trust boundary (allowing for coauthoring on documents protected by sensitivity labels)
- **Data_CppUncaughtExceptionCount:long** - Uncaught native exceptions while activity was running
- **Data_DetachedDuration:long** - Time for which Activity was detached/not running
- **Data_DstDoc_AccessMode:long** - How was this document opened (Read only | read write)
- **Data_DstDoc_AssistedReadingReasons:long** - Predefined set of values of why document was opened in assisted reading mode
- **Data_DstDoc_AsyncOpenKind:long** – Indicates whether a cached version of the new cloud document was opened and which asynchronous refresh logic was used.
- **Data_DstDoc_ChunkingType:long** - How is document stored in SharePoint
- **Data_DstDoc_EdpState:long** - Enterprise Data Protection state of document
- **Data_DstDoc_Ext:string** - Document extension
- **Data_DstDoc_Extension:string** - Document extension
- **Data_DstDoc_FileFormat:long** - Predefined set of values of format of file (more granular than extension)

- **Data_DstDoc_Fqdn:string** - Where is document stored (SharePoint.com, live.net), only available for Office 365 domains
- **Data_DstDoc_FqdnHash:string** - Hash of where document is stored
- **Data_DstDoc_IdentityTelemetryId:string** - Unique GUID of user
- **Data_DstDoc_IdentityUniqueId:string** - Unique identifier of identity that was used for Shared Documents action
- **Data_DstDoc_IOFlags:long** - Bitmask for various IO-related flags for a given document
- **Data_DstDoc_IrmRights:long** - Predefined set of values of what type of Information Rights Management is applied on this document (Forward, Reply, SecureReader, Edit etc.)
- **Data_DstDoc_IsCloudCollabEnabled:bool** - True if the "IsCloudCollabEnabled" HTTP header has already been received from an OPTIONS request.
- **Data_DstDoc_IsIncrementalOpen:bool** - Was document opened incrementally (new feature that opens document without needing to download entire document)
- **Data_DstDoc_IsOcsSupported:bool** - Is Document supports co-authoring using new OCS service
- **Data_DstDoc_IsOpeningOfflineCopy:bool** - verifies if document is being opened from local cache
- **Data_DstDoc_IsSyncBacked:bool** - Is document opened from folder that is using OneDrive sync back app
- **Data_DstDoc_Location:long** - Predefined set of values of where document is stored (Local, SharePoint, WOPI, Network etc.)
- **Data_DstDoc_LocationDetails:long** - Predefined set of values of more detailed location (Temp folder, downloads folder, One Drive Documents, One Drive Pictures etc.)
- **Data_DstDoc_NumberCoAuthors:long** - Number of co-authors at the time of opening of a document
- **Data_DstDoc_PasswordFlags:long** - Predefined set of values of how document is encrypted with password (None, password to read, password to edit)
- **Data_DstDoc_ReadOnlyReasons:long** - Predefined set of values of why this document was marked read only (Locked on server, final document, password protected to edit, etc.)
- **Data_DstDoc_ResourceIdHash:string** - Hash of resource identifier for documents stored in cloud
- **Data_DstDoc_ServerDocId:string** - immutable identifier for documents stored in cloud
- **Data_DstDoc_ServerProtocol:long** - Predefined set of values of which protocol is used to talk to server (Http, Cobalt, WOPI etc.)
- **Data_DstDoc_ServerType:long** - Predefined set of values of type of server (SharePoint, DropBox, WOPI)
- **Data_DstDoc_ServerVersion:long** - verifies if server is based off Office14, Office15, or Office 16
- **Data_DstDoc_SessionId:long** - generated GUID that Identifies the instance of the document within the same process session
- **Data_DstDoc_SharePointServiceContext:string** - An opaque string, typically GridManagerID.FarmID. Useful for correlating client side and server-side logs
- **Data_DstDoc_SizeInBytes:long** - Document size in bytes

- **Data_DstDoc_SpecialChars:long** - Bitmask indicating special chars in the document's URL or Path
- **Data_DstDoc_StorageProviderId:string** - A string that identifies the document's storage provider, like "DropBox"
- **Data_DstDoc_StreamAvailability:long** - Predefined set of values of status of document Stream(available, permanently disabled, not available)
- **Data_DstDoc_UrlHash:string** - hash of full URL of documents stored in cloud
- **Data_DstDoc_UsedWrsDataOnOpen:bool** - true if the file was opened incrementally using pre cached WRS data on the host
- **Data_DstDoc_WopiServiceId:string** - WOPI Service identifier, for example "Dropbox"
- **Data_FileType:long** - Predefined set of values of internal type of file
- **Data_fLifeguarded:bool** - Was document ever lifeguarded (feature to fix document errors by themselves without prompting user)?
- **Data_FWebCreated:bool** - Does this document have WebCreator flag?
- **Data_SaveReason:long** - Predefined set of values of why this save was performed? (AutoSave, ToOCSTransitionSave, ToCSITransitionSave, etc.)
- **Data_SaveType:long** - Predefined set of values of save type (SaveAs, Publish, Manual, OMSave, etc.)
- **Data_SrcDoc_AccessMode:long** - How was this document opened (Read only | read write)
- **Data_SrcDoc_AssistedReadingReasons:long** - Predefined set of values of why document was opened in assisted reading mode
- **Data_SrcDoc_AsyncOpenKind:long** – Indicates whether a cached version of the original cloud document was opened and which asynchronous refresh logic was used.
- **Data_SrcDoc_ChunkingType:long** - How is document stored in SharePoint
- **Data_SrcDoc_EdpState:long** - Enterprise Data Protection state of document
- **Data_SrcDoc_Ext:string** - Document extension
- **Data_SrcDoc_Extension:string** - Document extension
- **Data_SrcDoc_FileFormat:long** - Predefined set of values of format of file (more granular than extension)
- **Data_SrcDoc_Fqdn:string** - Where is document stored (SharePoint.com, live.net), only available for Office 365 domains
- **Data_SrcDoc_FqdnHash:string** - Hash of where document is stored
- **Data_SrcDoc_IdentityTelemetryId:string** - Unique GUID of user
- **Data_SrcDoc_IdentityUniqueId:string** - Unique identifier of identity that was used for Shared Documents action
- **Data_SrcDoc_IOFlags:long** - Bitmask for various IO-related flags for a given document
- **Data_SrcDoc_IrmRights:long** - Predefined set of values of what type of Information Rights Management is applied on this document (Forward, Reply, SecureReader, Edit etc.)
- **Data_SrcDoc_IsCloudCollabEnabled:bool** - True if the "IsCloudCollabEnabled" HTTP header has already been received from an OPTIONS request.

- **Data_SrcDoc_IsIncrementalOpen:bool** - Was document opened incrementally (new feature that opens document without needing to download entire document)
- **Data_SrcDoc_IsOcsSupported:bool** - Is Document supports co-authoring using new OCS service
- **Data_SrcDoc_IsOpeningOfflineCopy:bool** - verifies if document is being opened from local cache
- **Data_SrcDoc_IsSyncBacked:bool** - Is document opened from folder that is using OneDrive sync back app
- **Data_SrcDoc_Location:long** - Predefined set of values of where document is stored (Local, SharePoint, WOPI, Network etc.)
- **Data_SrcDoc_LocationDetails:long** - Predefined set of values of more detailed location (Temp folder, downloads folder, One Drive Documents, One Drive Pictures etc.)
- **Data_SrcDoc_NumberCoAuthors:long** - Number of co-authors at the time of opening of a document
- **Data_SrcDoc_PasswordFlags:long** - Predefined set of values of how document is encrypted with password (None, password to read, password to edit)
- **Data_SrcDoc_ReadOnlyReasons:long** - Predefined set of values of why this document was marked read only (Locked on server, final document, password protected to edit, etc.)
- **Data_SrcDoc_ResourceIdHash:string** - Hash of resource identifier for documents stored in cloud
- **Data_SrcDoc_ServerDocId:string** - immutable identifier for documents stored in cloud
- **Data_SrcDoc_ServerProtocol:long** - Predefined set of values of which protocol is used to talk to server (Http, Cobalt, WOPI etc.)
- **Data_SrcDoc_ServerType:long** - Predefined set of values of type of server (SharePoint, DropBox, WOPI)
- **Data_SrcDoc_ServerVersion:long** - verifies if server is based off Office14, Office15, or Office 16
- **Data_SrcDoc_SessionId:long** - generated GUID that identifies the instance of the document within the same process session
- **Data_SrcDoc_SharePointServiceContext:string** - An opaque string, typically GridManagerID.FarmID. Useful for correlating client side and server-side logs
- **Data_SrcDoc_SizeInBytes:long** - Document size in bytes
- **Data_SrcDoc_SpecialChars:long** - Bitmask indicating special chars in the document's URL or Path
- **Data_SrcDoc_StorageProviderId:string** - A string that identifies the document's storage provider, like "DropBox"
- **Data_SrcDoc_StreamAvailability:long** - Predefined set of values of status of document Stream(available, permanently disabled, not available)
- **Data_SrcDoc_UrlHash:string** - hash of full URL of documents stored in cloud
- **Data_SrcDoc_UsedWrsDataOnOpen:bool** - true if the file was opened incrementally using pre cached WRS data on the host
- **Data_SrcDoc_WopiServiceId:string** - WOPI Service identifier, for example "Dropbox"
- **Data_StopwatchDuration:long** - Total time for Activity
- **Data_TypeOfSaveDialog:long** - Predefined set of values of Dialog

(RUN_SAVEAS_DLG,RUN_SAVEMEDIA_DLG, RUN_SAVEAS_VIDEO_DLG etc.)

- **Data_WaitForSaveOrMergeSuccess:bool** - SaveAs succeeded waiting for a background save or merge.
- **Data_WaitForSaveOrMergeTimeout:long** - SaveAs timed out when waiting for a background save or merge.
- **DstDoc** - New location of document
- **SrcDoc** - Original location of document

Office.PowerPoint.DocOperation.SaveLegacy

Collected whenever PowerPoint performs a save using the legacy code path. Includes success or failure result type of save performance metrics and relevant document metadata. Failures in save can result in data loss. Microsoft uses this data to ensure the feature is working as expected and user content is successfully being persisted.

The following fields are collected:

- **Data_AddDocTelemetryResult:long** - Does this log entry have all necessary document telemetry (Data_Doc_* fields)? If not, why?
- **Data_ClpDocHasDrmDoc:bool** - Whether the document has a DRM document
- **Data_ClpDocHasIdentity:bool** - Whether the document has identity info (used to get and set sensitivity labels)
- **Data_ClpDocHasSessionMetadata:bool** – Whether the document has working sensitivity label metadata from the session
- **Data_ClpDocHasSpoMetadata:bool** - Whether the document has sensitivity label metadata from SPO via IMetadataCache
- **Data_ClpDocHasSpoPackage:bool** – Whether the document has sensitivity label metadata from SPO via IPackage
- **Data_ClpDocIsProtected:bool** - Whether or not the document is protected by IRM
- **Data_ClpDocMetadataSource:int** - Enum specifying where sensitivity label metadata is from (IRM, OPC part, Sharepoint etc)
- **Data_ClpDocNeedsUpconversion:bool** - Whether the document needs to upconvert sensitivity label data from the custom.xml part
- **Data_ClpDocNumFailedSetLabels:int** - Count of sensitivity labels that failed to set on the document
- **Data_ClpDocSessionMetadataDirty:bool** - Whether the document has working sensitivity label metadata that has been dirtied
- **Data_ClpDocWasInTrustBoundary:bool** – Whether the document was in the trust boundary (allowing for coauthoring on documents protected by sensitivity labels)
- **Data_CppUncaughtExceptionCount:long** - Uncaught native exceptions while activity was running
- **Data_DetachedDuration:long** - Time for which Activity was detached/not running
- **Data_Doc_AccessMode:long** - How was this document opened (Read only | read write)
- **Data_Doc_AssistedReadingReasons:long** - Predefined set of values of why document was opened in assisted reading mode

- **Data_Doc_AsyncOpenKind:long** – Indicates whether a cached version of the cloud document was opened and which asynchronous refresh logic was used.
- **Data_Doc_ChunkingType:long** - How is document stored in SharePoint
- **Data_Doc_EdpState:long** - Enterprise Data Protection state of document
- **Data_Doc_Ext:string** - Document extension
- **Data_Doc_Extension:string** - Document extension
- **Data_Doc_FileFormat:long** - Predefined set of values of format of file (more granular than extension)
- **Data_Doc_Fqdn:string** - Where is document stored (SharePoint.com, live.net), only available for Office 365 domains
- **Data_Doc_FqdnHash:string** - Hash of where document is stored
- **Data_Doc_IdentityTelemetryId:string** - Unique GUID of user
- **Data_Doc_IdentityUniqueId:string** - Unique identifier of identity that was used for Shared Documents action
- **Data_Doc_IOFlags:long** - Bitmask for various IO-related flags for a given document
- **Data_Doc_IrmRights:long** - Predefined set of values of what type of Information Rights Management is applied on this document (Forward, Reply, SecureReader, Edit etc.)
- **Data_Doc_IsCloudCollabEnabled:bool** - True if the "IsCloudCollabEnabled" HTTP header has already been received from an OPTIONS request.
- **Data_Doc_IsIncrementalOpen:bool** - Was document opened incrementally (new feature that opens document without needing to download entire document)
- **Data_Doc_IsOcsSupported:bool** - Is Document supports co-authoring using new OCS service
- **Data_Doc_IsOpeningOfflineCopy:bool** - verifies if document is being opened from local cache
- **Data_Doc_IsSyncBacked:bool** - Is document opened from folder that is using OneDrive sync back app
- **Data_Doc_Location:long** - Predefined set of values of where document is stored (Local, SharePoint, WOPI, Network etc.)
- **Data_Doc_LocationDetails:long** - Predefined set of values of more detailed location (Temp folder, downloads folder, One Drive Documents, One Drive Pictures etc.)
- **Data_Doc_NumberCoAuthors:long** - Number of co-authors at the time of opening of a document
- **Data_Doc_PasswordFlags:long** - Predefined set of values of how document is encrypted with password (None, password to read, password to edit)
- **Data_Doc_ReadOnlyReasons:long** - Predefined set of values of why this document was marked read only (Locked on server, final document, password protected to edit, etc.)
- **Data_Doc_ResourceIdHash:string** - Hash of resource identifier for documents stored in cloud
- **Data_Doc_RtcType** - Indicates how the real-time channel (RTC) was set up for current file (Disabled, unsupported, on demand, always on, etc.).
- **Data_Doc_ServerDocId:string** - immutable identifier for documents stored in cloud
- **Data_Doc_ServerProtocol:long** - Predefined set of values of which protocol is used to talk to server (Http, Cobalt, WOPI etc.)

- **Data_Doc_ServerType:long** - Predefined set of values of type of server (SharePoint, DropBox, WOPI)
- **Data_Doc_ServerVersion:long** - verifies if server is based off Office14, Office15, or Office 16
- **Data_Doc_SessionId:long** - generated GUID that Identifies the instance of the document within the same process session
- **Data_Doc_SharePointServiceContext:string** - An opaque string, typically GridManagerID.FarmID. Useful for correlating client side and server-side logs
- **Data_Doc_SizeInBytes:long** - Document size in bytes
- **Data_Doc_SpecialChars:long** - Bitmask indicating special chars in the document's URL or Path
- **Data_Doc_StorageProviderId:string** - A string that identifies the document's storage provider, like "DropBox"
- **Data_Doc_StreamAvailability:long** - Predefined set of values of status of document Stream(available, permanently disabled, not available)
- **Data_Doc_UrlHash:string** - hash of full URL of documents stored in cloud
- **Data_Doc_UsedWrsDataOnOpen:bool** - true if the file was opened incrementally using pre cached WRS data on the host
- **Data_Doc_WopiServiceId:string** - WOPI Service identifier, for example "Dropbox"
- **Data_DstDoc_AccessMode:long** - How was this document opened (Read only | read write)
- **Data_DstDoc_AssistedReadingReasons:long** - Predefined set of values of why document was opened in assisted reading mode
- **Data_DstDoc_AsyncOpenKind:long** – Indicates whether a cached version of the new cloud document was opened and which asynchronous refresh logic was used.
- **Data_DstDoc_ChunkingType:long** - How is document stored in SharePoint
- **Data_DstDoc_EdpState:long** - Enterprise Data Protection state of document
- **Data_DstDoc_Ext:string** - Document extension
- **Data_DstDoc_Extension:string** - Document extension
- **Data_DstDoc_FileFormat:long** - Predefined set of values of format of file (more granular than extension)
- **Data_DstDoc_Fqdn:string** - Where is document stored (SharePoint.com, live.net), only available for Office 365 domains
- **Data_DstDoc_FqdnHash:string** - Hash of where document is stored
- **Data_DstDoc_IdentityTelemetryId:string** - Unique GUID of user
- **Data_DstDoc_IdentityUniqueId:string** - Unique identifier of identity that was used for Shared Documents action
- **Data_DstDoc_IOFlags:long** - Bitmask for various IO-related flags for a given document
- **Data_DstDoc_IrmRights:long** - Predefined set of values of what type of Information Rights Management is applied on this document (Forward, Reply, SecureReader, Edit etc.)
- **Data_DstDoc_IsCloudCollabEnabled:bool** - True if the "IsCloudCollabEnabled" HTTP header has already been received from an OPTIONS request.

- **Data_DstDoc_IsIncrementalOpen:bool** - Was document opened incrementally (new feature that opens document without needing to download entire document)
- **Data_DstDoc_IsOcsSupported:bool** - Is Document supports co-authoring using new OCS service
- **Data_DstDoc_IsOpeningOfflineCopy:bool** - verifies if document is being opened from local cache
- **Data_DstDoc_IsSyncBacked:bool** - Is document opened from folder that is using OneDrive sync back app
- **Data_DstDoc_Location:long** - Predefined set of values of where document is stored (Local, SharePoint, WOPI, Network etc.)
- **Data_DstDoc_LocationDetails:long** - Predefined set of values of more detailed location (Temp folder, downloads folder, One Drive Documents, One Drive Pictures etc.)
- **Data_DstDoc_NumberCoAuthors:long** - Number of co-authors at the time of opening of a document
- **Data_DstDoc_PasswordFlags:long** - Predefined set of values of how document is encrypted with password (None, password to read, password to edit)
- **Data_DstDoc_ReadOnlyReasons:long** - Predefined set of values of why this document was marked read only (Locked on server, final document, password protected to edit etc.)
- **Data_DstDoc_ResourceIdHash:string** - Hash of resource identifier for documents stored in cloud
- **Data_DstDoc_ServerDocId:string** - immutable identifier for documents stored in cloud
- **Data_DstDoc_ServerProtocol:long** - Predefined set of values of which protocol is used to talk to server (Http, Cobalt, WOPI etc.)
- **Data_DstDoc_ServerType:long** - Predefined set of values of type of server (SharePoint, DropBox, WOPI)
- **Data_DstDoc_ServerVersion:long** - verifies if server is based off Office14, Office15, or Office 16
- **Data_DstDoc_SessionId:long** - generated GUID that Identifies the instance of the document within the same process session
- **Data_DstDoc_SharePointServiceContext:string** - An opaque string, typically GridManagerID.FarmID. Useful for correlating client side and server-side logs
- **Data_DstDoc_SizeInBytes:long** - Document size in bytes
- **Data_DstDoc_SpecialChars:long** - Bitmask indicating special chars in the document's URL or Path
- **Data_DstDoc_StorageProviderId:string** - A string that identifies the document's storage provider, like "DropBox"
- **Data_DstDoc_StreamAvailability:long** - Predefined set of values of status of document Stream(available, permanently disabled, not available)
- **Data_DstDoc_UrlHash:string** - hash of full URL of documents stored in cloud
- **Data_DstDoc_UsedWrsDataOnOpen:bool** - true if the file was opened incrementally using pre cached WRS data on the host
- **Data_DstDoc_WopiServiceId:string** - WOPI Service identifier, for example "Dropbox"
- **Data_FileType:long** - Predefined set of values of internal type of file
- **Data_fLifeguarded:bool** - Was document ever lifeguarded (feature to fix document errors by

themselves without prompting user)?

- **Data_SaveReason:long** - Predefined set of values of why this save was performed? (AutoSave, ToOCSTransitionSave, ToCSITransitionSave etc.)
- **Data_SaveType:long** - Predefined set of values of save type (SaveAs, Publish, Manual, OMSave etc.)
- **Data_SrcDoc_AccessMode:long** - How was this document opened (Read only | read write)
- **Data_SrcDoc_AssistedReadingReasons:long** - Predefined set of values of why document was opened in assisted reading mode
- **Data_SrcDoc_AsyncOpenKind:long** – Indicates whether a cached version of the original cloud document was opened and which asynchronous refresh logic was used.
- **Data_SrcDoc_ChunkingType:long** - How is document stored in SharePoint
- **Data_SrcDoc_EdpState:long** - Enterprise Data Protection state of document
- **Data_SrcDoc_Ext:string** - Document extension
- **Data_SrcDoc_Extension:string** - Document extension
- **Data_SrcDoc_FileFormat:long** - Predefined set of values of format of file (more granular than extension)
- **Data_SrcDoc_Fqdn:string** - Where is document stored (SharePoint.com, live.net), only available for Office 365 domains
- **Data_SrcDoc_FqdnHash:string** - Hash of where document is stored
- **Data_SrcDoc_IdentityTelemetryId:string** - Unique GUID of user
- **Data_SrcDoc_IdentityUniqueId:string** - Unique identifier of identity that was used for Shared Documents action
- **Data_SrcDoc_IOFlags:long** - Bitmask for various IO-related flags for a given document
- **Data_SrcDoc_IrmRights:long** - Predefined set of values of what type of Information Rights Management is applied on this document (Forward, Reply, SecureReader, Edit etc.)
- **Data_SrcDoc_IsCloudCollabEnabled:bool** - True if the "IsCloudCollabEnabled" HTTP header has already been received from an OPTIONS request.
- **Data_SrcDoc_IsIncrementalOpen:bool** - Was document opened incrementally (new feature that opens document without needing to download entire document)
- **Data_SrcDoc_IsOcsSupported:bool** - Is Document supports co-authoring using new OCS service
- **Data_SrcDoc_IsOpeningOfflineCopy:bool** - verifies if document is being opened from local cache
- **Data_SrcDoc_IsSyncBacked:bool** - Is document opened from folder that is using OneDrive sync back app
- **Data_SrcDoc_Location:long** - Predefined set of values of where document is stored (Local, SharePoint, WOPI, Network etc.)
- **Data_SrcDoc_LocationDetails:long** - Predefined set of values of more detailed location (Temp folder, downloads folder, One Drive Documents, One Drive Pictures, etc.)
- **Data_SrcDoc_NumberCoAuthors:long** - Number of co-authors at the time of opening of a document
- **Data_SrcDoc_PasswordFlags:long** - Predefined set of values of how document is encrypted with

password (None, password to read, password to edit)

- **Data_SrcDoc_ReadOnlyReasons:long** - Predefined set of values of why this document was marked read only (Locked on server, final document, password protected to edit, etc.)
- **Data_SrcDoc_ResourceIdHash:string** - Hash of resource identifier for documents stored in cloud
- **Data_SrcDoc_ServerDocId:string** - immutable identifier for documents stored in cloud
- **Data_SrcDoc_ServerProtocol:long** - Predefined set of values of which protocol is used to talk to server (Http, Cobalt, WOPI etc.)
- **Data_SrcDoc_ServerType:long** - Predefined set of values of type of server (SharePoint, DropBox, WOPI)
- **Data_SrcDoc_ServerVersion:long** - verifies if server is based off Office14, Office15, or Office 16
- **Data_SrcDoc_SessionId:long** - generated GUID that Identifies the instance of the document within the same process session
- **Data_SrcDoc_SharePointServiceContext:string** - An opaque string, typically GridManagerID.FarmID. Useful for correlating client side and server-side logs
- **Data_SrcDoc_SizeInBytes:long** - Document size in bytes
- **Data_SrcDoc_SpecialChars:long** - Bitmask indicating special chars in the document's URL or Path
- **Data_SrcDoc_StorageProviderId:string** - A string that identifies the document's storage provider, like "DropBox"
- **Data_SrcDoc_StreamAvailability:long** - Predefined set of values of status of document Stream(available, permanently disabled, not available)
- **Data_SrcDoc_UrlHash:string** - hash of full URL of documents stored in cloud
- **Data_SrcDoc_UsedWrsDataOnOpen:bool** - true if the file was opened incrementally using pre cached WRS data on the host
- **Data_SrcDoc_WopiServiceId:string** - WOPI Service identifier, for example "Dropbox"
- **Data_StopwatchDuration:long** - Total time for Activity
- **Data_TypeOfSaveDialog:long** - Predefined set of values of Dialog (RUN_SAVEAS_DLG, RUN_SAVEMEDIA_DLG, RUN_SAVEAS_VIDEO_DLG etc.)
- **Doc** - current document for Save
- **DstDoc** - New location of document (in case of SaveAs)
- **SrcDoc** - Original location of document (in case of SaveAs)

Office.PowerPoint.PPT.Android.RehearseView.FeedbackReceived

This event is used to analyze the seen/tried/kept funnel of the feature. This event along with Seen and Tried event helps us determine if users are dropping out of the funnel. The data is used to determine into if users are dropping because of errors faced during the experience. This helps us maintain the health of the feature.

The following fields are collected:

- None

Office.PowerPoint.PPT.Android.RehearseView.SessionStarted

This event is used to analyze the seen/tried/kept funnel of the feature. This event along with Seen and Kept event helps us figure out if users are dropping out of the funnel. This data is used to understand if users are

dropping because of errors faced during experience. This helps us maintain the health of the feature.

The following fields are collected:

- None

Office.PowerPoint.PPT.IOS.RehearseView

This event is denoting that user has stopped the rehearsal session. The data is used in combination with Office.PowerPoint.IOS.Android.RehearseView.StartSession as the first indicator of any crashes or errors that user faces.

The following fields are collected:

- **ConnectionCreationTime** - time taken to create service side connections.
- **CountDownAlertTime** - Time for which countdown alert was displayed.
- **CountdownInitTime** – Time between slideshow load completed and countdown started.
- **CritiqueSummary** - Summary of what all critiques user saw with their counts.
- **ExitEventCode** - Code to identify under which scenario user exit out of rehearse session, whether it was error scenario or successful exit
- **FRETime** - Time between FRE screen started to display until user dismissed it.
- **MicrophonePermissionTime** - Time for which microphone permission alert was displayed until user selected one of the options.
- **PauseRehearsingCount** - Count of how many times user clicked on pause rehearsal
- **RehearsalInitTime** - Time taken by rehearsal to initialize
- **ResumeRehearsingCount** - Count of how many times user clicked on resume rehearsal
- **Sessionid** - This is speech front door session ID. This is used to debug service logs.
- **SlideshowViewLoadTime** - Time taken by slideshow to load.

Office.PowerPoint.PPT.IOS.RehearseView.RehearsalSummaryPage

Event is triggered when summary page has finished loading. This event helps us capture the performance of the summary page. It tells how much time it takes for the rehearsal summary service page to load on the client. It is required to keep the feature performant.

The following fields are collected:

- **PayloadCreationTime** - This is the time taken in milliseconds to create payload.
- **PostUrlCallTime** - This is the time taken in milliseconds to send the post URL call.
- **RehearseSessionId** - This is speech front door session ID. We can use this to debug service logs.
- **SummaryPageErrorReceived** - This is a Boolean value that indicates if summary page was received or error occurred.
- **SummaryPageHtmlLoadTime** - This is the time taken in milliseconds to load summarypageHtml.
- **SummaryPageLoadStartTime** - This is the time taken in milliseconds receive first response from the server.
- **SummaryPageLoadTime** - Time taken to load summary page. This includes payload creation time.
- **ThumbnailsCount** - This is the total number of thumbnails that will be part of summary page.

Office.PowerPoint.PPT.IOS.RehearseView.StartSession

This event is triggered when the user clicks on start session. This event helps us capture how many users are using the Presenter coach feature on iOS. When combined with Office.PowerPoint.PPT.iOS.RehearseView it will tell us how many users successfully completed the rehearsal session and how many couldn't. This is our first indicator of crashes or errors in the feature.

The following fields are collected:

- None

Office.PowerPoint.PPT.Mac.Shell.PrintInfo

Collected whenever an export PDF operation has completed and contains information about the success of the operation. This information is critical to identify the success of export PDF operations for our application.

The following fields are collected:

- Data_ExportAsPDFSucceed - Boolean indicating if exporting as PDF was a success.

Office.PowerPoint.PPT.Shared.RehearseView.RehearseClicked

This event captures when RehearseWithCoach is clicked. This event is used to analyze the seen- tried-kept funnel of the feature. This event along with tried and kept event helps us figure out if users are dropping out of the funnel. This helps us maintain the health of the feature.

The following fields are collected:

- None

Office.PowerPoint.PPT.Shared.SlideShow.Failure

Collecting failures during slide show as a key feature for PowerPoint. Microsoft is collecting when error happens during slide show to help improve user experience on slide show. Microsoft uses this data to get diagnostic information about where the error happens while user is using slide show.

The following fields are collected:

- CountOArtErrors - total number of OArt errors
- CountOtherErrors - total number of other errors
- CountPPTErrors - total number of PPT errors
- CountSlideShowErrors - total number of slide show errors
- FirstOArtError - first error happened in OArt
- FirstOtherError - first error happened in other area
- FirstPPTError - first error happened in PPT
- FirstSlideShowError - first error happened in slide show

Office.PowerPoint.Rehearsal.RehearseAgain

This event captures when Rehearse Again is clicked in the summary report. This event is used to analyze the success of the PowerPoint Coach entry points, and is an indicator of the health of the feature.

The following fields are collected:

- None

Office.PowerPoint.RunPrintOperation

Collected whenever a print PDF operation has completed and contains information about the layout type, use of slide numbers, and the success of the operation. This information is critical to identify the success of print PDF operations for our application.

The following fields are collected:

- **Data_PrintWithSlideNumbers** - Boolean indicating if the user is printing with slide numbers.
- **Data_SavePrintLayoutType** - The print layout type at the time of starting the print or export operation.
- **Data_Success** - Boolean indicating if printing was a success.

Office.Project.ProjectFileSave

Project saves a file. This event indicates a Project save. It allows Microsoft to measure success of Project saving a file, which is important to avoid document data loss.

The following fields are collected:

- **Data_TriggerTime** - time of save
- **Data_FileType** - type of file the project is being saved as

Office.Session.Activity.Start

Allows us to know when a data streamer session has started. Used for feature health and monitoring. This event is generated by Microsoft Data Streamer for Excel Add-in.

The following fields are collected:

- **Activity_Name** - Name of the activity "Session"
- **Activity_CV** - ID to correlate the events across the connection session
- **Activity_StartStopType** - Start
- **Activity_DateTimeTicks** - Data Time for the activity

Office.Session.Activity.Stop

Allows us to know when a data streamer session has stopped Used for feature health and monitoring. This event is generated by Microsoft Data Streamer for Excel Add-in.

The following fields are collected:

- **Activity_Name** - Name of the activity "Session"
- **Activity_CV** - ID to correlate the events across the connection session
- **Activity_StartStopType** - Stop
- **Activity_DateTimeTicks** - Data Time for the activity

Office.StreamDevice.Activity.Start

Allows us to know if start streaming data source is successful. Used for feature health and monitoring. This event is generated by Microsoft Data Streamer for Excel Add-in.

The following fields are collected:

- **Datasource_Type** - Serial device, or App Service information
- **DataSource_Name** - Name of connected data source
- **Activity_Name** - Name of the activity "StreamDeviceData" or "StreamFileData"
- **Activity_CV** - ID to correlate the events across the connection session
- **Activity_StartStopType** - Start
- **Activity_DateTimeTicks** - Data Time for the activity

Office.StreamDevice.Activity.Stop

Allows us to know if stop streaming data source is successful. Used for feature health and monitoring. This event is generated by Microsoft Data Streamer for Excel Add-in.

The following fields are collected:

- **Datasource_Type** - Serial device, or App Service information
- **DataSource_Name** - Name of connected data source
- **Activity_Name** - Name of the activity "StreamDeviceData" or "StreamFileData"
- **Activity_CV** - ID to correlate the events across the connection session
- **Activity_StartStopType** - Stop
- **Activity_DateTimeTicks** - Data Time for the activity

Office.TargetedMessaging.ABExperimentMessageTrigger

Tracks the number of users receiving BizBar and Dynamic Canvas messages from TargetedMessagingService (TMS). This data is critical to understand what messages users are getting and on what surface so that we can ensure they are not missing any messages that may be critical to their continued use of the product. Also needed to accurately measure the success of our experiments and campaigns run through TMS.

The following fields are collected:

- **Data_Surface** – Name of the surface that this service delivered message is meant for
- **Data_Flight** – ECS/CT Flight identifier that was used to deliver this message
- **Data_CampaignId** – identifier of the campaign that this message is part of
- **Data_MessageId** – identifier of this service delivered message
- **Data_TransactionId** – identifier of this transaction with the service
- **Data_TriggerPoint** – The step in which this event was logged (message received vs message displayed)

Office.Text.FontPickerFontSelected.Win32

This event indicates whether the downloaded font was rendered correctly. Used to indicate success or failure of Font Download.

The following fields are collected:

- **Font name (Data_Font)** - name of font picked in font picker
- **User click (Data_FClick)** - if user used mouse to select item

Office.Text.ResourceClient.RequestResourceInternal

This event indicates whether the font was downloaded correctly. Used to indicate success or failure of rendering the downloaded font.

The following fields are collected:

- **Data_FontToken** - resource file name will be saved as
- **Data_HTTPResult** - result of the HTTP request
- **Data_HTTPStatusCode** - HTTP code returned from the HTTP request
- **Data_isInternetOn** - If we had connection when trying to retrieve the resource
- **Data_RequestUrl** - URL of the CDN resource we're trying to retrieve

Office.Translator.DocumentTranslated

Collects success or failure of a full document translation a user trigger in the Translator SDX. This is critical to evaluate the health of the translation feature and react to any outages that might occur. Monitor the health of the "Translate Document" scenario in Word.

The following fields are collected:

- **Data.actionSource** - How was the translate selection triggered
- **Data.bodyBackgroundColor** - Office theme container background color
- **Data.bodyForegroundColor** - Office theme container foreground color
- **Data.browserLang** - Browser current display language
- **Data.browserOnline** - Obsolete
- **Data.browserPlatform** - Browser platform
- **Data.browserUserAgent** - Browser user agent
- **Data.colorDepth** - System color depth
- **Data.contentLanguage** - Detected language of the content to translate
- **Data.controlBackgroundColor** - Office theme control background color
- **Data.controlForegroundColor** - Office theme control foreground color
- **Data.correlationId** - Unique identifier of the request sent to the service
- **Data.crossSessionId** - Unique identifier of the user
- **Data.crossSessionStartTime** - UTC timestamp of when the translation session started
- **Data.currentTime** - UTC timestamp of when this telemetry message was sent
- **Data.displayLanguage** - Office display language
- **Data.documentSourceLang** - Document content language
- **Data.documentTargetLang** - Language document will be translated to
- **Data.environment** - Service environment the request is sent to
- **Data.errorMessage** - Error message reported by the service
- **Data.eventActionType** - Type of telemetry event
- **Data.eventTagId** - Unique identifier of the line of code that produced this telemetry message.
- **Data.flights** - Enabled flights
- **Data.fileSize** - Size of Word file to translate
- **Data.fileSource** - Where is the Word file hosted (offline, online)
- **Data.fileType** - Word file extension
- **Data.innerHeight"** - Side pane container height
- **Data.innerWidth"** - Side pane container width
- **Data.lookupSourceLang** - Not used for document translation
- **Data.lookupTargetLang** - Not used for document translation

- **Data.officeHost** - Office application hosting the side pane
- **Data.officeLocale** - Office user language
- **Data.officeMachineId** - Device Unique identifier
- **Data.officePlatform** - Device platform
- **Data.officeSessionId** - Office session identifier
- **Data.officeUserId** - Office user unique identifier
- **Data.officeVersion** - Office version
- **Data.pageXOffset** - Side pane horizontal scroll position from the left side of the pane
- **Data.pageYOffset** - Side pane vertical scroll position from the top side of the pane
- **Data.pixelDepth** - Screen color resolution
- **Data.responseCode** - Request response code from the service
- **Data.responseTime** - Request elapsed time
- **Data.resultType** - Request result
- **Data.screenHeight** - Screen height in pixels
- **Data.screenLeft** - Horizontal coordinate of the window relative to the screen
- **Data.screenTop** - Vertical coordinate of the window relative to the screen
- **Data.screenWidth** - Screen width in pixels
- **Data.selectedTab** - Which tab is selected Selection or Document
- **Data.serverUrl** - Translation service URL
- **Data.sessionId** - Side pane session identifier
- **Data.sessionStartTime** - UTC Timestamp of when the translation session started
- **Data.sourceTextHash** - Hash of text to translate
- **Data.sourceTextLength** - Text to translate length
- **Data.sourceTextWords** - Number of words in the text to translate
- **Data.warningMessage** - Warning message reported by the service

Office.Translator.TextTranslated

Collects success or failure of a selection translation that a user action triggers in the Translator SDX. This is critical to evaluate the health of the translation feature and react to any outages that might occur. Used to monitor the health of the "Translate Selection" scenario in Excel, PowerPoint, Word.

The following fields are collected:

- **Data.actionSource** - How was the translate selection triggered
- **Data.bodyBackgroundColor** - Office theme container background color
- **Data.bodyForegroundColor** - Office theme container foreground color
- **Data.browserLang** - Browser current display language
- **Data.browserOnline** - Obsolete

- **Data.browserPlatform** - Browser platform
- **Data.browserUserAgent** - Browser user agent
- **Data.colorDepth** - System color depth
- **Data.contentLanguage** - Detected language of the content to translate
- **Data.controlBackgroundColor** - Office theme control background color
- **Data.controlForegroundColor** - Office theme control foreground color
- **Data.correlationId** - Unique identifier of the request sent to the service
- **Data.crossSessionId** - Unique identifier of the user
- **Data.crossSessionStartTime** - UTC timestamp of when the translation session started
- **Data.currentTime** - UTC timestamp of when this telemetry message was sent
- **Data.displayLanguage** - Office display language
- **Data.documentSourceLang** - Not used for selection
- **Data.documentTargetLang** - Nor used for translation selection
- **Data.environment** - Service environment the request is sent to
- **Data.errorMessage** - Error message reported by the service
- **Data.eventActionType** - Type of telemetry event
- **Data.eventTagId"** - Unique identifier of the line of code that produced this telemetry message
- **Data.flights** - Enabled flights
- **Data.innerHeight** - Side pane container height
- **Data.innerWidth** - Side pane container width
- **Data.lookupSourceLang** - Language of the currently selected text
- **Data.lookupTargetLang** - Language currently selected text will be translated to
- **Data.officeHost** - Office application hosting the side pane
- **Data.officeLocale** - Office user language
- **Data.officeMachineId** - Device Unique identifier
- **Data.officePlatform** - Device platform
- **Data.officeSessionId** - Office session identifier
- **Data.officeUserId** - Office user unique identifier
- **Data.officeVersion** - Office version
- **Data.pageXOffset** - Side pane horizontal scroll position from the left side of the pane
- **Data.pageYOffset** - Side pane vertical scroll position from the top side of the pane
- **Data.pixelDepth** - Screen color resolution
- **Data.responseCode** - Request response code from the service

- **Data.responseTime** - Request elapsed time
- **Data.resultType** - Request result
- **Data.screenHeight** - Screen height in pixels
- **Data.screenLeft** - Horizontal coordinate of the window relative to the screen
- **Data.screenTop** - Vertical coordinate of the window relative to the screen
- **Data.screenWidth** - Screen width in pixels
- **Data.selectedTab** - Which tab is selected Selection or Document
- **Data.serverUrl** - Translation service URL
- **Data.sessionId** - Side pane session identifier
- **Data.sessionStartTime** - UTC Timestamp of when the translation session started
- **Data.sourceTextHash** - Hash of text to translate
- **Data.sourceTextLength** - Text to translate length
- **Data.sourceTextWords** - Number of words in the text to translate
- **Data.warningMessage** - Warning message reported by the service

Office.UX.AccChecker.AccCheckerFinalViolationCountPerRule

This event is triggered when Accessibility issues are reported for the currently opened document. This event represents the Accessibility violations (Errors, Warnings, and Tips) that exist per Rule, for the opened document at the beginning and end of the session. This event is used to record the counts of Accessibility violations (Errors, Warnings, and Tips) per Rule, for the opened document at the beginning and end of the session.

Details of violation counts per rule help Microsoft to identify which Accessibility issues are most common in Office documents. This helps with work on remediating them and drives the creation of an inclusive environment in the workplace and classroom for people with disabilities.

The following fields are collected:

- **Data_FinalCount_RuleID_0** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_1** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_2** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_3** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_4** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_5** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_6** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_7** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.

- **Data_FinalCount_RuleID_8** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_9** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_10** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_11** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_12** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_13** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_14** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_15** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_16** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_FinalCount_RuleID_17** - Number of violations of Rule ID = n that remain the last time the acc checker ran in a session.
- **Data_InitialCount_RuleID_0** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_1** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_2** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_3** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_4** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_5** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_6** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_7** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_8** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_9** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.

- **Data_InitialCount_RuleID_10** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_11** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_12** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_13** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_14** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_15** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_16** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **Data_InitialCount_RuleID_17** - Number of violations of Rule ID = n that were found the first time the acc checker ran in a session.
- **FinalDocID** - Final DocumentID of the scanned document
- **FinalDocUrlHash** - Final DocumentURLHash of the scanned document
- **InitialDocID** - Initial DocumentID of the scanned document
- **InitialDocUrlHash** - Initial DocumentURLHash of the scanned document
- **PaneOpened** - A boolean that tracks whether the AccChecker pane was opened
- **ServerDocID** - Server DocumentID for the document scanned by the Accessibility checker

Office.UX.AccChecker.AccCheckerViolationInformation

This event is triggered when Accessibility issues are reported for the currently opened document. It represents the aggregate counts of violations (Errors, Warnings, and Tips) for the opened document at the beginning and end of the session. This event is used to record the aggregate counts of Accessibility violations (Errors, Warnings, and Tips) for the opened document at the beginning and end of the session. The knowledge of Accessibility checker usage allows Microsoft to improve its application experiences to be more inclusive for people with disabilities in Office usage scenarios for the workplace and classroom.

The following fields are collected:

- **FinalDocID** - Final DocumentID of the scanned document
- **FinalDocUrlHash** - Final DocumentURLHash of the scanned document
- **FinalErrorCount** - Final count of Errors reported by Accessibility Checker for the document
- **FinalIntelligentServiceCount** - Final count of Intelligent services issues reported by Accessibility Checker for the document
- **FinalTipCount** - Final count of tips reported by Accessibility Checker for the document
- **FinalViolationCount** - Final count of violations reported by Accessibility Checker for the document
- **FinalWarningCount** - Final count of warnings reported by Accessibility Checker in the document
- **InitialDocID** - Initial DocumentID of the scanned document

- **InitialDocUrlHash** - Initial DocumentURLHash of the scanned document
- **InitialErrorCount** - Number of all violations of type Error that were found the first time the acc checker ran in a session.
- **InitialIntelligentServicesCount** - Number of all violations of type Intelligent Service that were found the first time the acc checker ran in a session.
- **InitialTipCount** - Number of all violations of type Tip that were found the first time the acc checker ran in a session.
- **InitialUrlHash** - Number of all violations of type error that were found the first time the acc checker ran in a session.
- **InitialViolationCount** - Number of all violations that were found the first time the acc checker ran in a session.
- **InitialWarningCount** - Number of all violations of type Warning that were found the first time the acc checker ran in a session.
- **PaneOpened** - A Boolean that tracks whether the Accessibility Checker pane was opened
- **ServerDocID** - Server DocumentID for the document scanned by the Accessibility checker

Office.UX.AccChecker.BackgroundAccCheckerEnabledState

This event is triggered when the user or IT Admin has enabled the Background Accessibility checker for the Office user. This event is used to understand the instances when the Background Accessibility checker is enabled for Office users. The enabled status of the Background Accessibility checker allows Microsoft to understand if documents can be scanned automatically in the background. This helps to create a more inclusive workplace and classroom environment for people with disabilities.

The following fields are collected:

- **BackgroundAccCheckerEnabled** - Boolean to track the Enabled/Disabled state of the Background Accessibility checker

Office.UX.AccChecker.BackgroundScanningCheckboxClicked

This event is triggered when the user enables the Background Accessibility checker from the Accessibility checker task pane. This event is used to understand the instances when the Background Accessibility checker is enabled for Office documents. The enabled status of the Background Accessibility checker allows Microsoft to understand if documents can be scanned automatically in the background. This helps to create a more inclusive workplace and classroom environment for people with disabilities.

The following fields are collected:

- **FinalBackgroundScanningState** - Initial state of the checkbox that enables background scanning
- **InitialBackgroundScanningState** - Initial state of the checkbox that enables background scanning

Office.UX.AccChecker.DisabledResults

This event is triggered when the Accessibility Checker gets disabled for the opened document. This event is used to understand the instances when the Office Accessibility checker gets disabled, due to a legacy or unsupported Office document. The disabled status of the Accessibility checker allows Microsoft to understand how often a document cannot be scanned, and to assist users in allowing the scanning of such documents – by upconverting the document to a modern file format. This helps to create a more inclusive workplace and classroom environment for people with disabilities.

The following fields are collected:

- **Data_Disabled_ID** - ID of the disabled error

- **Data_Disabled_Reason** - Reason for disabling the Accessibility Checker
- **Data_IsUpConvertEnabled** - Tracks whether Upconvert to a modern file format is available for the document

Office.UX.AccChecker.ShowTaskPane

This event is triggered when the Accessibility Checker task pane is launched for the currently opened document. This event is used to understand the usage of the Office Accessibility Checker. The Accessibility Checker is used to identify and remediate Accessibility issues in Office documents. The knowledge of Accessibility Checker usage allows Microsoft to improve its application experiences to be more inclusive for people with disabilities in Office usage scenarios for the workplace and classroom.

The following fields are collected:

- **BackgroundScanCheckboxEnabled** - Tracks whether the Background Accessibility Checker is enabled
- **Column** - Purpose
- **DocUrlHash** - Unique doc ID hash of the document that was scanned
- **HasAccessibilityViolations** - Tracks if the document contains any accessibility violations at the point the pane is opened
- **IsPaneDisabled** - Tracks if the Accessibility Checker pane is opened in a disabled state (legacy or unsupported document)
- **PaneOpenedBefore** - Tracks if the Accessibility Checker pane was opened before
- **WAC_ServerDocId** - Server Document ID for the document that was scanned

Office.Visio.Shared.FeatureExperimentation

Tracks feature flighting for users. This event helps us determines success or failure of feature flights.

The following fields are collected:

- **Data_Enable:bool**- true indicate the feature is enabled for current user
- **Data_Feature:string** - name of the feature
- **Data_Flighted:bool** - true indicates the feature is enabled
- **Data_Licensed:bool** - true indicates the feature is under licensing check
- **Data_Subscriber:bool** - true indicates the user has subscription license

Office.Visio.Shared.RefreshSmartDiagram

Captures diagram refresh failures when file is created through DV. This helps us debug the failures and issues in data refresh in a DV diagram.

The following fields are collected:

- **Data_ConnectorsBasedOnSequence:bool** - true if the refreshed diagram was originally created using connector based on sequence" option
- **Data_DialogError:string** - error during refreshing smart diagram
- **Data_FileError:string** - error string when connected Excel file is invalid
- **Data_OverwriteSelected:bool** - true if user selected overwrite diagram option during refresh
- **Data_WarningShown:bool** - true if there was any warning shown to user during data refresh

Office.Visio.Shared.WritebackToExcel

Captures Excel write back failures when file is created through DV. This helps us debug the failures and issues in writing back data to Excel in a DV diagram.

The following fields are collected:

- **Data_ConnectorsBasedOnSequence:bool** - true means connectors are created based on sequence settings
- **Data_DataSourceType:string** - This filed indicates whether diagram is created from "Table" or "CustomRange"
- **Data_DialogError:string** - Custom Error type while creating smart diagram through Excel
- **Data_NoOfShapesAdded:int** - Number of shapes added during writeback to Excel functionality
- **Data_NoOfShapesDeleted:int** - Number of shapes deleted during writeback to Excel functionality
- **Data_OverwriteSelected:bool** - true indicate user selected overwrite data option
- **Data_SourceDataModified:bool** - true indicates source data is modified
- **Data_WarningShown:bool** - true means data update warning shown to the user
- **Data_WarningShownBecauseOfPresenceOfFormula:bool** - true indicates warning shown to the user because of presence of formula in Excel
- **Data_WarningShownToAddNextStepID:bool** - true indicates warning show to the user because next step Identifier missing in Excel
- **Data_WarningShownToConvertToTable:bool** - true indicates warning shown to the user to convert Excel data to table format

Office.Word.FileNew.CreateNewFile

This event indicates that a new document is created in Office Word and tracks success or failure of the operation. The event is used to monitor that new document creation is working as expected. It is also used to calculated monthly active users/devices and cloud reliability metrics.

The following fields are collected:

- **Data_DirtyState** - whether the document was created in a dirty state (with changes that need to be saved)
- **Data_ErrorID** - error identifier in case of operation failure
- **Data_MainPdod** - The document identifier during this process session
- **Data_UsesCustomTemplate** - indicates whether the document was created from a custom template

Office.Word.FileOpen.UserInitiatedOpen

This event indicates Office Word opens a document by user initiation instead of by Office Word programmatically. It also contains critical file open performance data and is an app start event from user perspective. The event monitors whether file-open is working as expected. It is also used to calculated monthly active users/devices, and cloud reliability metrics.

The following fields are collected:

- **Data_AddDocTelemRes** - Reports whether we were able to properly populate other document telemetry-related values in the event. Used for data quality diagnostics.
- **Data_BytesAsynchronous** - Number of bytes (compressed) that we believe we can open the file without if we get them before the user wants to start editing or maybe saving.

- **Data_BytesAsynchronousWithWork** - Number of bytes (compressed) that we might be able to open the file without but would require significant code investments to make it happen
- **Data_BytesSynchronous** - Number of bytes (compressed) that we must have before we can start opening the file
- **Data_BytesUnknown** - Number of bytes in document parts that we don't expect to find.
- **Data_Doc_AccessMode** - Document is read only/editable
- **Data_Doc_AssistedReadingReasons** - Predefined set of values of why document was opened in assisted reading mode
- **Data_Doc_ChunkingType** - Units used for incremental document open
- **Data_Doc_EdpState** - Electronic Data Protection setting for the document
- **Data_Doc_Ext** - Document extension (docx/xlsb/pptx, etc.)
- **Data_Doc_FileFormat** - File format protocol version
- **Data_Doc_Fqdn** - OneDrive or SharePoint Online Domain Name
- **Data_Doc_FqdnHash** - One-way hash of customer identifiable domain name
- **Data_Doc_IdentityTelemetryId** - A one-way hash of the user identity used to perform the open
- **Data_Doc_InitializationScenario** - Records how the document was opened
- **Data_Doc_IOFlags** - Reports on the cached flags used to set open request options
- **Data_Doc_IrmRights** - Actions permitted by the Electronic Data Protection policy that has been applied to the document/user
- **Data_Doc_IsIncrementalOpen** - Flag indicating that the document has been incrementally opened
- **Data_Doc_IsOcsSupported** - Flag indicating that the document is supported in the collaboration service
- **Data_Doc_IsOpeningOfflineCopy** - Flag indicating that the offline copy of a document was opened
- **Data_Doc_IsSyncBacked** - Flag indicating that an auto synced copy of the document exists on the computer
- **Data_Doc_Location** - Indicates which service provided the document (OneDrive, File Server, SharePoint)
- **Data_Doc_LocationDetails** - Indicates which Known Folder provided a locally stored document
- **Data_Doc_NumberCoAuthors** - Count of the number of fellow users in a collaborative editing session
- **Data_Doc_PasswordFlags** - Indicates read or read/write password flags set
- **Data_Doc_ReadOnlyReasons** - Reasons why the document was opened read only
- **Data_Doc_ResourceIdHash** - An anonymized document Identifier used to diagnose problems
- **Data_Doc_ServerDocId** - An immutable anonymized document Identifier used to diagnose problems
- **Data_Doc_ServerProtocol** - the protocol version used to communicate with the service
- **Data_Doc_ServerType** - the type of the server offering the service (SharePoint, OneDrive, WOPI etc.)
- **Data_Doc_ServerVersion** - the server version offering the service

- **Data_Doc_SessionId** - the server version offering the service
- **Data_Doc_SharePointServiceContext** - Diagnostic information from SharePoint Online requests
- **Data_Doc_SizeInBytes** - Indicator of document size
- **Data_Doc_SpecialChars** - Indicator of special chars in the document's URL or Path
- **Data_Doc_StreamAvailability** - Indicator if document stream is available/disabled
- **Data_Doc_SyncBackedType** - Indicator as to the type of document (local or service based)
- **Data_Doc_UrlHash** - One-way hash to create a naïve document identifier
- **Data_Doc_WopiServiceId** - Contains unique identifier of WOPI service provider
- **Data_EditorDisablingRename** - Identifier of the first editor that caused for rename to be disabled
- **Data_EditorsCount** - Number of editors in the document
- **Data_ForceReadWriteReason** - Integer value representing the reason why the file was forced into read/write mode
- **Data_FSucceededAfterRecoverableFailure** - Indicates that open succeeded after repairing a failure while opening the document
- **Data_LastLoggedTag** - Unique tag for code call site used to identify when we try to fail the open twice (used for data quality diagnostics)
- **Data_LinkStyles** - Indicates whether we are linking to the template styles
- **Data_MainPdod** - The document identifier in Office Word process
- **Data_Measurements** - Encoded string containing the time breakdown of the different parts of open. Used to diagnose open performance.
- **Data_MoveDisabledReason** - Error that is disabling move for the document
- **Data_MoveFlightEnabled** - Whether the flight for the move feature is enabled
- **Data_OpenInitiateKind** – Type of the scenario where users started this file-open operation.
- **Data_PartsUnknown** - The number of document parts that we couldn't get data for
- **Data_RecoverableFailureInitiationLocationTag** - Unique tag for code call site used to identify the place in code where we attempt to fix the file before opening it
- **Data_RenameDisabledReason** - Error that is causing for rename to be disabled for this document
- **Data_RenameFlightEnabled** - Whether the flight for the rename feature is enabled
- **Data_SecondaryTag** - Unique tag for code call site used to add additional failure data for open.
- **Data_TemplateFormat** - File format of the template that the document is based on.
- **Data_UsesNormal** - Indicates whether the open document is based on the normal template.
- **Data_VerboseMeasurements** - Encoded string containing the detailed time breakdown of the different parts of open. Used to measure performance, only enabled for internal rings.

Office.Word.FileSave.ActCmdGosubSaveAs

This event indicates that a user is saving their changes to a new document. The event monitors whether saving to a new document is working as expected. It is also used to calculated monthly active users/devices and cloud reliability metrics.

The following fields are collected:

- **Data_AddDocTelemRes** - Reports whether we were able to properly populate other document telemetry-related values in the event. Used for data quality diagnostics.
- **Data_DetachedDuration** - How long was the activity detached from the thread
- **Data_Doc_AccessMode** - Document is read only/editable
- **Data_Doc_AssistedReadingReasons** - Predefined set of values of why document was opened in assisted reading mode
- **Data_Doc_AsyncOpenKind** – Indicates whether a cached version of the cloud document was opened and which asynchronous refresh logic was used.
- **Data_Doc_ChunkingType** - Units used for incremental document open
- **Data_Doc_EdpState** - Electronic Data Protection setting for the document
- **Data_Doc_Ext** - Document extension (docx/xlsb/pptx etc.)
- **Data_Doc_FileFormat** - File format protocol version
- **Data_Doc_Fqdn** - OneDrive or SharePoint Online Domain Name
- **Data_Doc_FqdnHash** - One-way hash of customer identifiable domain name
- **Data_Doc_IdentityTelemetryId** - A one-way hash of the user identity used to perform the open
- **Data_Doc_InitializationScenario** - Records how the document was opened
- **Data_Doc_IOFlags** - Reports on the cached flags used to set open request options
- **Data_Doc_IrmRights** - Actions permitted by the Electronic Data Protection policy that has been applied to the document/user
- **Data_Doc_IsIncrementalOpen** - Flag indicating that the document has been incrementally opened
- **Data_Doc_IsOcsSupported** - Flag indicating that the document is supported in the collaboration service
- **Data_Doc_IsOpeningOfflineCopy** - Flag indicating that the offline copy of a document was opened
- **Data_Doc_IsSyncBacked** - Flag indicating that an auto synced copy of the document exists on the computer
- **Data_Doc_Location** - Indicates which service provided the document (OneDrive, File Server, SharePoint etc.)
- **Data_Doc_LocationDetails** - Indicates which Known Folder provided a locally stored document
- **Data_Doc_NumberCoAuthors** - Count of the number of fellow users in a collaborative editing session
- **Data_Doc_PasswordFlags** - Indicates read or read/write password flags set
- **Data_Doc_ReadOnlyReasons** - Reasons why the document was opened read only
- **Data_Doc_ResourceIdHash** - An anonymized document identifier used to diagnose problems
- **Data_Doc_RtcType** - Indicates how the real-time channel (RTC) was set up for current file (Disabled, unsupported, on demand, always on, etc.).
- **Data_Doc_ServerDocId** - An immutable anonymized document identifier used to diagnose problems

- **Data_Doc_ServerProtocol** - The protocol version used to communicate with the service
- **Data_Doc_ServerType** - The type of the server offering the service (SharePoint, OneDrive, WOPI etc.)
- **Data_Doc_ServerVersion** - The server version offering the service
- **Data_Doc_SessionId** - Identifies a specific document edit session within the full session
- **Data_Doc_SharePointServiceContext** - Diagnostic information from SharePoint Online requests
- **Data_Doc_SizeInBytes** - Indicator of document size
- **Data_Doc_SpecialChars** - Indicator of special chars in the document's URL or Path
- **Data_Doc_StreamAvailability** - Indicator if document stream is available/disabled
- **Data_Doc_SyncBackedType** - Indicator as to the type of document (local or service based)
- **Data_Doc_UrlHash** - One-way hash to create a naïve document identifier
- **Data_EditorDisablingRename** - identifier of the first editor that caused rename to be disabled
- **Data_EditorsCount** - Number of editors in the document
- **Data_LastLoggedTag** - Unique tag for code call site used to identify when we fail to try the save twice (used for data quality diagnostics)
- **Data_MoveDisabledReason** - Error that is disabling move for the document
- **Data_MoveFlightEnabled** - Whether the flight for the move feature is enabled
- **Data_RenameDisabledReason** - Error that is causing rename to be disabled for the document
- **Data_RenameFlightEnabled** - Whether the flight for the rename feature is enabled

Office.Word.FileSave.ActFConfirmSaveDocCoreQuerySave

This event indicates Office Word prompts the user to save changes when it tries to close the document. It allows Microsoft to monitor whether save-at-quit works as expected to avoid document data loss. The event monitors whether save-at-quit is working as expected. It is also used to calculated monthly active users/devices and cloud reliability metrics.

The following fields are collected:

- **Data_AddDocTelemRes** - Reports whether we were able to properly populate other document telemetry-related values in the event. Used for data quality diagnostics.
- **Data_DetachedDuration** - How long was the activity detached from the thread
- **Data_Doc_AccessMode** - Document is read only/editable
- **Data_Doc_AssistedReadingReasons** - Predefined set of values of why document was opened in assisted reading mode
- **Data_Doc_AsyncOpenKind** – Indicates whether a cached version of the cloud document was opened and which asynchronous refresh logic was used.
- **Data_Doc_ChunkingType** - Units used for incremental document open
- **Data_Doc_EdpState** - Electronic Data Protection setting for the document
- **Data_Doc_Ext** - Document extension (docx/xlsb/pptx etc.)
- **Data_Doc_FileFormat** - File format protocol version

- **Data_Doc_Fqdn** - OneDrive or SharePoint Online Domain Name
- **Data_Doc_FqdnHash** - One-way hash of customer identifiable domain name
- **Data_Doc_IdentityTelemetryId** - A one-way hash of the user identity used to perform the open
- **Data_Doc_InitializationScenario** - Records how the document was opened
- **Data_Doc_IOFlags** - Reports on the cached flags used to set open request options
- **Data_Doc_IrmRights** - Actions permitted by the Electronic Data Protection policy that has been applied to the document/user
- **Data_Doc_IsIncrementalOpen** - Flag indicating that the document has been incrementally opened
- **Data_Doc_IsOcsSupported** - Flag indicating that the document is supported in the collaboration service
- **Data_Doc_IsOpeningOfflineCopy** - Flag indicating that the offline copy of a document was opened
- **Data_Doc_IsSyncBacked** - Flag indicating that an auto synced copy of the document exists on the computer
- **Data_Doc_Location** - Indicates which service provided the document (OneDrive, File Server, SharePoint etc.)
- **Data_Doc_LocationDetails** - Indicates which Known Folder provided a locally stored document
- **Data_Doc_NumberCoAuthors** - Count of the number of fellow users in a collaborative editing session
- **Data_Doc_PasswordFlags** - Indicates read or read/write password flags set
- **Data_Doc_ReadOnlyReasons** - Reasons why the document was opened read only
- **Data_Doc_ResourceIdHash** - An anonymized document identifier used to diagnose problems
- **Data_Doc_RtcType** - Indicates how the real-time channel (RTC) was set up for current file (Disabled, unsupported, on demand, always on, etc.).
- **Data_Doc_ServerDocId** - An immutable anonymized document identifier used to diagnose problems
- **Data_Doc_ServerProtocol** - the protocol version used to communicate with the service
- **Data_Doc_ServerType** - the type of the server offering the service (SharePoint, OneDrive, WOPI etc.)
- **Data_Doc_ServerVersion** - the server version offering the service
- **Data_Doc_SessionId** - Identifies a specific document edit session within the full session
- **Data_Doc_SharePointServiceContext** - Diagnostic information from SharePoint Online requests
- **Data_Doc_SizeInBytes** - Indicator of document size
- **Data_Doc_SpecialChars** - Indicator of special chars in the document's URL or Path
- **Data_Doc_StreamAvailability** - Indicator if document stream is available/disabled
- **Data_Doc_SyncBackedType** - Indicator as to the type of document (local or service based)
- **Data_Doc_UrlHash** - One-way hash to create a naïve document identifier
- **Data_Doc_WopiServiceId** - Contains unique identifier of WOPI service provider
- **Data_DstDoc_AccessMode** - Destination Document is read only/editable

- **Data_DstDoc_EdpState** -Electronic Data Protection setting for the destination document-
- **Data_DstDoc_Ext** - Document extension (docx/xlsb/pptx, etc.) for the destination document
- **Data_DstDoc_FileFormat** - File format protocol version for the destination document
- **Data_DstDoc_Location** - Indicates which service will provide storage for destination document (OneDrive, File Server, SharePoint etc.)
- **Data_DstDoc_LocationDetails** - Indicates which local Known Folder stored the destination document
- **Data_DstDoc_SessionId** - Identifies a specific document edit session within the full session
- **Data_DstDoc_UrlHash** - One-way hash to create a naïve document identifier for the destination document
- **Data_FailureClass** - Integer representing the failure class for OCS transition failures
- **Data_LocationPickerSaveStatus** - Integer value that indicates the action that triggered the save from the save on exit dialog
- **Data_MainPdod** - The document identifier in Office Word process.
- **Data_MoveFlightEnabled** - Whether the flight for the move feature is enabled
- **Data_OCSSyncbackSaveStarted** - Flag that indicates that this save is related to sync back save
- **Data_RenameDisabledReason** - Error that is causing for rename to be disabled for this document
- **Data_RenameFlightEnabled** - Whether the flight for the rename feature is enabled
- **Data_SaveInitiateKind** - Integer that indicates how the save was initiated
- **Data_SrcDocIsUnnamedOrNew** - Indicates whether the document we are saving is new

Office.Word.FileSave.SaveAsSaveFile

This event indicates Office Word saves a document into a new document. It allows Microsoft to detect errors in save-as which is important to avoid document data loss. The event monitors whether save-as is working as expected. It is also used to calculated monthly active users/devices and cloud reliability metrics.

The following fields are collected:

- **Data_AddDocTelemRes** - Reports whether we were able to properly populate other document telemetry-related values in the event. Used for data quality diagnostics.
- **Data_AddDocTelemResDst** - Reports whether we were able to properly populate other document telemetry-related values in the event for the destination document. Used for data quality diagnostics.
- **Data_AddDocTelemResSrc** - Reports whether we were able to properly populate other document telemetry-related values in the event for the source document. Used for data quality diagnostics.
- **Data_DetachedDuration** - How long was the activity detached from the thread
- **Data_Doc_AccessMode** - Document is read only/editable
- **Data_Doc_AssistedReadingReasons** - Predefined set of values of why document was opened in assisted reading mode
- **Data_Doc_AsyncOpenKind** – Indicates whether a cached version of the cloud document was opened and which asynchronous refresh logic was used.
- **Data_Doc_ChunkingType** - Units used for incremental document open

- **Data_Doc_EdpState** - Electronic Data Protection setting for the document
- **Data_Doc_Ext** - Document extension (docx/xlsb/pptx etc.)
- **Data_Doc_FileFormat** - File format protocol version
- **Data_Doc_Fqdn** - OneDrive or SharePoint Online Domain Name
- **Data_Doc_FqdnHash** - One-way hash of customer identifiable domain name
- **Data_Doc_IdentityTelemetryId** - A one-way hash of the user identity used to perform the open
- **Data_Doc_IOFlags** - Reports on the cached flags used to set open request options
- **Data_Doc_IrmRights** - Actions permitted by the Electronic Data Protection policy that has been applied to the document/user
- **Data_Doc_IsIncrementalOpen** - Flag indicating that the document has been incrementally opened
- **Data_Doc_IsOcsSupported** - Flag indicating that the document is supported in the collaboration service
- **Data_Doc_IsOpeningOfflineCopy** - Flag indicating that the offline copy of a document was opened
- **Data_Doc_IsSyncBacked** - Flag indicating that an auto synced copy of the document exists on the computer
- **Data_Doc_Location** - Indicates which service provided the document (OneDrive, File Server, SharePoint etc.)
- **Data_Doc_LocationDetails** - Indicates which Known Folder provided a locally stored document
- **Data_Doc_NumberCoAuthors** - Count of the number of fellow users in a collaborative editing session
- **Data_Doc_ReadOnlyReasons** - Reasons why the document was opened read only
- **Data_Doc_ResourceIdHash** - An anonymized document identifier used to diagnose problems
- **Data_Doc_RtcType** - Indicates how the real-time channel (RTC) was set up for current file (Disabled, unsupported, on demand, always on, etc.).
- **Data_Doc_ServerDocId** - An immutable anonymized document identifier used to diagnose problems
- **Data_Doc_ServerProtocol** - the protocol version used to communicate with the service
- **Data_Doc_ServerType** - the type of the server offering the service (SharePoint, OneDrive, WOPI etc.)
- **Data_Doc_ServerVersion** - the server version offering the service
- **Data_Doc_SessionId** - Identifies a specific document edit session within the full session
- **Data_Doc_SharePointServiceContext** - Diagnostic information from SharePoint Online requests
- **Data_Doc_SizeInBytes** - Indicator of document size
- **Data_Doc_SpecialChars** - Indicator of special chars in the document's URL or Path
- **Data_Doc_StreamAvailability** - Indicator if document stream is available/disabled
- **Data_Doc_UrlHash** - One-way hash to create a naïve document identifier
- **Data_DstDoc_AccessMode** - Destination Document is read only/editable
- **Data_DstDoc_AssistedReadingReasons** - Predefined set of values of why the destination document was opened in assisted reading mode

- **Data_DstDoc_AsyncOpenKind** – Indicates whether a cached version of the new cloud document was opened and which asynchronous refresh logic was used.
- **Data_DstDoc_ChunkingType** - Units used for incremental document open
- **Data_DstDoc_EdpState** - Electronic Data Protection setting for the destination document
- **Data_DstDoc_Ext** - Document extension (docx/xlsb/pptx etc.)
- **Data_DstDoc_FileFormat** - File format protocol version
- **Data_DstDoc_Fqdn** - OneDrive or SharePoint Online Domain Name for the destination document
- **Data_DstDoc_FqdnHash** - One-way hash of customer identifiable domain name for the destination document
- **Data_DstDoc_IdentityTelemetryId** - A one-way hash of the user identity used to perform the open
- **Data_DstDoc_InitializationScenario** - Records how the destination document was opened
- **Data_DstDoc_IOFlags** - Reports on the cached flags used to set open request options for the destination document
- **Data_DstDoc_IrmRights** - Actions permitted by the Electronic Data Protection policy that has been applied to the destination document/user
- **Data_DstDoc_IsIncrementalOpen** - Flag indicating that the document has been incrementally opened
- **Data_DstDoc_IsOcsSupported** - Flag indicating that the document is supported in the collaboration service
- **Data_DstDoc_IsOpeningOfflineCopy** - Flag indicating that the offline copy of a document was opened
- **Data_DstDoc_IsSyncBacked** - Flag indicating that an auto synced copy of the document exists on the computer
- **Data_DstDoc_Location** - Indicates which service provided the storage for the destination document (OneDrive, File Server, SharePoint etc.)
- **Data_DstDoc_LocationDetails** - Indicates which Known Folder provided a locally stored document
- **Data_DstDoc_NumberCoAuthors** - Count of the number of fellow users in a collaborative editing session
- **Data_DstDoc_PasswordFlags** - Indicates read or read/write password flags set for the destination document
- **Data_DstDoc_ReadOnlyReasons** - Reasons why the destination document was opened read only
- **Data_DstDoc_ResourceIdHash** - An anonymized document identifier used to diagnose problems
- **Data_DstDoc_ServerDocId** - An immutable anonymized document identifier used to diagnose problems
- **Data_DstDoc_ServerProtocol** - the protocol version used to communicate with the service
- **Data_DstDoc_ServerType** - the type of the server offering the service (SharePoint, OneDrive, WOPI etc.)
- **Data_DstDoc_ServerVersion** - the server version offering the service
- **Data_DstDoc_SessionId** - Identifies a specific document edit session within the full session

- **Data_DstDoc_SharePointServiceContext** - Diagnostic information from SharePoint Online requests
- **Data_DstDoc_SizeInBytes** - Indicator of document size
- **Data_DstDoc_SpecialChars** - Indicator of special chars in the document's URL or Path
- **Data_DstDoc_StreamAvailability** - Indicator if document stream is available/disabled
- **Data_DstDoc_SyncBackedType** - Indicator as to the type of document (local or service based)
- **Data_DstDoc_UrlHash** - One-way hash to create a naïve document identifier for the destination document
- **Data_DstDoc_WopiServiceId** - Contains unique identifier of WOPI service provider
- **Data_FailureClass** - Integer representing the failure class for OCS transition failures
- **Data_LocationPickerPropagateToSaveTime,spLapsedMsec** - Measure the time, in milliseconds, that it takes for the save to trigger after getting a result from the location picker
- **Data_LocationPickerSaveStatus** - Status returned by the location picker
- **Data_MainPdod** - The document identifier in Office Word process
- **Data_MoveDisabledReason** - Error that is disabling move for the document
- **Data_MoveFlightEnabled** - Whether the flight for the move feature is enabled
- **Data_RenameDisabledReason** - Error that is causing for rename to be disabled for this document
- **Data_RenameFlightEnabled** - Whether the flight for the rename feature is enabled
- **Data_SaveInitiateKind** - Integer that indicates how the save was initiated
- **Data_SrcDoc_AccessMode** - Source Document is read only/editable
- **Data_SrcDoc_AssistedReadingReasons** - Predefined set of values of why document was opened in assisted reading mode
- **Data_SrcDoc_AsyncOpenKind** – Indicates whether a cached version of the original cloud document was opened and which asynchronous refresh logic was used.
- **Data_SrcDoc_ChunkingType** - Units used for incremental document open
- **Data_SrcDoc_EdpState** - Electronic Data Protection setting for the source document
- **Data_SrcDoc_Ext** - Document extension for the source document (docx/xlsb/pptx, etc.)
- **Data_SrcDoc_FileFormat** - File format protocol version for the source document
- **Data_SrcDoc_Fqdn** - OneDrive or SharePoint Online Domain Name for the source document
- **Data_SrcDoc_FqdnHash** - One-way hash of customer identifiable domain name for the source document
- **Data_SrcDoc_IdentityTelemetryId** - A one-way hash of the user identity used to perform the open
- **Data_SrcDoc_InitializationScenario** - Records how the document was opened
- **Data_SrcDoc_IOFlags** - Reports on the cached flags used to set open request options
- **Data_SrcDoc_IrmRights** - Actions permitted by the Electronic Data Protection policy that has been applied to the document/user
- **Data_SrcDoc_IsIncrementalOpen** - Flag indicating that the document has been incrementally opened

- **Data_SrcDoc_IsOcsSupported** - Flag indicating that the document is supported in the collaboration service
- **Data_SrcDoc_IsOpeningOfflineCopy** - Flag indicating that the offline copy of a document was opened
- **Data_SrcDoc_IsSyncBacked** - Flag indicating that an auto synced copy of the document exists on the computer
- **Data_SrcDoc_Location** - Indicates which service provided the source document (OneDrive, File Server, SharePoint, etc.)
- **Data_SrcDoc_LocationDetails** - Indicates which Known Folder provided a locally stored document
- **Data_SrcDoc_NumberCoAuthors** - Count of the number of fellow users in a collaborative editing session
- **Data_SrcDoc_PasswordFlags** - Indicates read or read/write password flags set
- **Data_SrcDoc_ReadOnlyReasons** - Reasons why the document was opened read only
- **Data_SrcDoc_ResourceIdHash** - An anonymized document identifier used to diagnose problems
- **Data_SrcDoc_ServerDocId** - An immutable anonymized document identifier used to diagnose problems
- **Data_SrcDoc_ServerProtocol** - the protocol version used to communicate with the service
- **Data_SrcDoc_ServerType** - the type of the server offering the service (SharePoint, OneDrive, WOPI etc.)
- **Data_SrcDoc_ServerVersion** - the server version offering the service
- **Data_SrcDoc_SessionId** - Identifies a specific document edit session within the full session
- **Data_SrcDoc_SharePointServiceContext** - Diagnostic information from SharePoint Online requests
- **Data_SrcDoc_SizeInBytes** - Indicator of document size
- **Data_SrcDoc_SpecialChars** - Indicator of special chars in the document's URL or Path
- **Data_SrcDoc_StreamAvailability** - Indicator if document stream is available/disabled
- **Data_SrcDoc_SyncBackedType** - Indicator as to the type of document (local or service based)
- **Data_SrcDoc_UrlHash** - One-way hash to create a naïve document identifier
- **Data_SrcDoc_WopiServiceId** - Contains unique identifier of WOPI service provider
- **Data_SrcDocIsUnnamedOrNew** - Indicates whether the document we are saving is new

Office.Word.Word.DocumentDirtyFlagChanged

This event indicates Office Word edits a document that changes the document internal state into "dirty". It allows Microsoft to evaluate the feature health of edit-document. The event is a heartbeat of user edits. It is also used to calculated monthly active users/devices.

The following fields are collected:

- **Data_CollectionTime**- Timestamp of the event
- **Data_DocumentLocation**- type of the document location
- **Data_DocumentLocationDetails**- Sub-type of the document location

- **Data_FAlwaysSaveEnabled**- Indicates whether always-save was enabled
- **Data_FirstEditTime**- Timestamp of first edit
- **Data_NumberCoAuthors**- Number of co-authors editing the document during the session
- **Data_NumberOfTimesDocumentDirtied**- Number of edits made to the document
- **Data_Pdod**- Document identifier in Office Word process
- **Data_UrlHash**- Hash of the document path
- **Data_ViewKind**- Type of Word view

OneNote.App.Navigation.RatingReminderDialogShown

The critical signal used to measure effectiveness of trigger logic for Rating reminder. This dialog is shown when the user has met all the conditions to see the rating reminder (no. of active days, has rated previously or not, etc.). This is used to ensure that the trigger logic for Rating reminder. If the users are seeing this dialog, it will provide us with ways to receive feedback from the customers on the right time and improve app health.

The following fields are collected:

- None

ParseLicenseOp

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when parsing licenses operation is performed.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name
- **AppInfo.Version** - Application version
- **iKey** - Logger server ID
- **RMS.ApplicationScenarioId** - Scenario ID provided by the application
- **RMS.Duration** - Total time for the operation to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the operation
- **RMS.HttpCall** - indicate if there is HTTP operation
- **RMS.LicenseFormat** - The license Format: Xrml or Json
- **RMS.Result** - Success or fail of the operation
- **RMS.ScenarioId** - Scenario ID defined by Rights Management Service Client
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.ServerType** - The type of Rights Management Service Server
- **RMS.StatusCode** - Status code of the operation result
- **RMS.VerifyCertChainDuration** - Duration time to verify certificate chain

- **RMS.VerifySignatureDuration** - Duration time to verify signature

qr.code.scan

This event lets us know when a user signs into Outlook Mobile by scanning an auth QR code on a desktop Outlook client which securely contains the user's sign-in information, thereby eliminating the need for manual sign-in. The event is used to detect the successful initiation and completion of the user authentication process using QR functionality. The event diagnoses sign-in errors that could prevent the user from successfully authenticating in the mobile app.

The following fields are collected:

- **action** - what action has the user taken for the qrcode flow

read.conversation

This event is triggered when an email is not visible on the device screen anymore. Used for monitoring possible negative impact on the health and performance of rendering an email message.

The following fields are collected:

- **above_40fps** - count of frames rendered above 40 fps
- **above_50fps** - count of frames rendered above 50 fps
- **above_55fps** - count of frames rendered above 55 fps
- **adal_id** - the account's active directory authentication ID, a unique identifier in the Microsoft authentication system
- **component_name** - the name of the component/view that is active during the filtering
- **event_mode** - the place in the app that the user entered the conversation (groups or other)
- **internet_message_id** - a tracking ID for the most recent message in the conversation
- **orientation** - the screen orientation at the time of the even (portrait or landscape)
- **recent_message_id** - the ID of the most recent message in the conversation
- **start_time** - timestamp of when the email message was visible to user.
- **suggested_reply_state** - the state of suggested replies for this conversation (unavailable, available, shown, used, or discarded)
- **suggested_reply_types** - indicates type and count of suggested reply shown/used for this conversation. It's a dictionary. For example, {text: 2, send_avail: 1}.
- **total_count** - total frames displayed by the component
- **view_duration** - how long the component was viewed by the user

save.attempt

Allows us to identify the impact of issues caused by users attempting to save a file by evaluating the number of sessions impacted and if there are common features of those sessions.

The following fields are collected:

- **file_type** - The type of file the user tried to save (such as .doc)
- **origin** - Where the file save attempt originated from (such as from an email) so we can detect issues associated with saving a file from a specific place in the app
- **token_type** - the type of token used to authenticate the account in order to save the file to help us detect authentication issues associated with saving a file

search.subtab.selected

The event collects origin points for the reason a search sub_tab was selected. The sub tabs sit under the primary app search bar to filter data. This event lets us track the entity type pills (all, mail, contacts, and calendar) that users are using when they do their searches so we can ensure the search filter mechanisms are working properly.

The following fields are collected:

- **properties_general** -The general properties that all Aria event is collecting
- **selected_reason** - The cause of the type pill getting selected, which could be one of the following values (glyph being an icon): tap_on_header, tap_on_see_all, enter_search_mode, mail_glyph, calendar_glyph.
- **subtab_type** - The type pill that got selected, which could be one of these four values: all, mail, contact, event.

send.message

Data collected indicates possible negative impact on the performance and health of sending email messages. The data is used to understand if feature is functioning successfully and to plan feature improvement for images in emails.

The following fields are collected:

- **account** - tracks the account that performed the action
- **compose_addressing_duration** - indicates the total time user spends on To/Cc/Bcc fields
- **compose_duration** - tracks the total time user took to compose the message including multiple drafts session
- **draft_message_id** - tracks the compose message ID of the message being sent
- **event_mode** - tracks the event mode if applicable to the message ("groups" or "other")
- **from_message_reminder** - Indicates if the message was sent in response to a message reminder
- **has_attachment** - indicates whether message has any attachments
- **has_mip_label** - indicates whether a MIP label was stamped on the message or not
- **image_attachment_count** - indicates how many images are being sent as attachments to the message
- **image_body_count** - indicates how many images are being sent inline as part of the body of the message
- **image_movement_count** - indicates how many images of the message that have been moved to inline or move back.
- **is_group_escalation** - is this a group escalated message, "escalated message" is a message that was sent to the user's mailbox because of an escalation (subscribed to group)
- **is_groups** - track whether message sent is a groups message or not
- **key_stroke_count** - tracks the keystrokes count for the message that is being sent
- **message_id** - tracks the message ID being replied/forwarded
- **origin** - indicates where compose was initiated, that is, new, reply, quick reply etc.
- **send_draft_origin** - indicates where send was initiated, that is, compose or quick reply
- **smart_compose_model_version** - tracks which version of smart compose model is being used

- **source_inbox** - indicates source inbox type for reference message,
- **suggested_reply_state** - capturing suggested reply state that is, unavailable, available, shown, used, discarded for this sent mail
- **suggested_reply_types** - indicates type and count of suggested reply shown/used for this sent email. It's a dictionary. For example, {text: 2, send_avail: 1}.
- **suggestions_requested** - indicates how many smart compose suggestions requested
- **suggestions_results** - smart compose suggestions' result, i.e accepted, rejected
- **suggestions_returned** - indicates how many smart compose suggestions returned from server
- **suggestions_shown** - indicates how many smart compose suggestions shown to the user
- **thread_id** - indicates thread ID of the conversation being replied/forwarded

session

Allows us to detect and fix situations where we are using up too much of your device's battery and helps us identify what could be the cause.

The following fields are collected:

- **battery_level** - tells us the battery level on the device to help us detect when our app is causing a negative impact on your device's battery level
- **has_hx** - Tells us the account is using our new sync service to help us detect issues caused by our sync service
- **Session.Duration** - the length of the session in seconds
- **Session.DurationBucket** - duration length time bucket *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **Session.FirstLaunchTime** - first recorded launch time of the app *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **Session.State** - the indicator of whether this is the start or end of the session

settings.action

This event collects the configuration information in settings. The data allows us to detect situations where there is possible negative impact on the users' ability configure app settings, such as notification settings, primary mail account, and configuring the email signature.

The following fields are collected:

- **account_order_changed** - To check if you changed the order of your accounts to make sure this configuration works properly
- **action** - possible actions taken in settings, such as deleting an account to help us diagnose issues and ensure no negative impact
- **auth_type** - The authentication type being used by the account, so we understand which backend sync layer we are using to help us diagnose issues
- **changed_folder** - Capturing whether a folder was changed to help us diagnose issues.
- **delete_scope** - During an account deletion, whether you deleted the account from this device or from all devices with Outlook.
- **emo_default_provider_selected_type** - Field that determines the type of the default meeting provider

set by the user.

- **emo_default_provider_switch_type** - The type of switch done by the user between the online meeting providers in the Every Meeting Online screen. Helps us to understand the user's engagement with the feature.
- **enabled_state** - Whether your auto reply, save contacts, and block external images settings are configured correctly
- **notification_action** - To check if you have configured any notification actions for triaging emails to help us make sure this setting is working successfully
- **notification_action_number** - To check if your notification actions (action one or action two) are configured correctly
- **server_type** - Similar to auth_type, it tells us which type of account you have to help us diagnose issues better. Examples: Office365, Gmail, Outlook
- **setting_properties** - Tracks properties relation to setting action detailed below:
  - **alternate_app_icon_setting** - the selected alternate app icon (light, dark)
  - **auth_type** - indicates the back-end authentication type allowing us to know if there is an issue with a particular account type
  - **badge_count_state** - indicates what type of badge count the user has asked for that is, no badges, focused inbox only, etc.
  - **changed_folder** - determines whether this action was archived, scheduled, or another action.
  - **delete_scope** - tracks whether this action was related to deleting someone just on this device or on all devices, if applicable.
  - **enabled_state** - whether state related to the action is enabled
  - **in_app_language** - the selected in-app language, string type (default, en-US, fa, ru etc.)
  - **notification_action_setting** - indicates the details of, if applicable, notification action settings related to this action
    - **notification_action** - indicates what the user was trying to do, that is, flag, delete, archive, it allows us to determine what message action the user wanted to perform on the notification and if the action failed or not.
    - **notification_action_number** - indicates which action number (two of three actions are customizable) was assigned a notification action, that is, action one, action two. This allows us to determine if there is a problem with a particular action.
  - **notification_state** - indicates what type of badge count the user has asked for that is, no badges, focused inbox only, etc.
  - **server_type** - indicates the back-end server type allowing us to know if there is an issue with a particular server type
  - **source** - indicates what is the source of notifications, if applicable, from settings or do not disturb setting
  - **swipe_setting** - indicates the details of, if applicable, swipe settings related to this action
    - **swipe_action** - indicates what the user was trying to do, that is, flag, delete, archive, it allows us to determine what action the user wanted and if the action failed or not.
    - **swipe_direction** - indicates which way the user set up the swipe to be, that is, left to right or right to left. This allows us to determine if there is a problem with a particular swipe direction.
  - **temperature_unit_setting** - the selected temperature unit to be used for weather
  - **theme_color_setting** - the custom app theme color selected by the user
  - **ui_mode_setting** - the selected UI mode (dark, light, system default, low battery etc.)
  - **signature_setting** - indicates if the setting was applied to all account or an individual account

- **state_changed_to** - To check if your focused inbox On/Off setting is configured correctly
- **swipe_action** - To check if you have configured any swipe actions for triaging emails to help us make sure this setting is working successfully
- **swipe_direction** - To check if your swipe directions (left or right) are configured correctly

sidebar.action

Allows us to detect situations where there is possible negative impact on your ability configure your app settings, such as your notification settings, your primary mail account, and configuring your mail signature.

Data fields that are common for Outlook Mobile for this event on iOS and Android:

- **Account** - tracks the account and its data associated with the event, values tracked in this data are in the common om field documentation *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **action** - tracks the type of side bar action occurred, that is, dismissed, help button selected, mail side bar, etc.,
- **from_favorites** - tracks if the action is coming from an item in favorites
- **mail_folder_type** - what type of folder was selected during the side bar action, if any.
- **sidebar_type** - tracks the type of side bar associated with this event, that is, mail or calendar to help us ensure the navigation from the favorites setting works properly

The following fields are collected:

- **account_type** - indicates what authentication type the account is that is, Gmail, outlook, etc.
- **account_has_groups** - Helps us make sure if the account has groups, they are configured correctly
- **calendar_accounts_count** - Number of calendar accounts you have to help us make sure your calendar accounts are configured correctly
- **calendar_apps_count** - Number of calendar apps you have to help us make sure your interesting calendar apps are configured correctly
- **calendar_type** - The type of calendar you have (Primary calendar, Group calendar, etc.)
- **has_favorite_folders** - Helps us make sure favorite folders are configured correctly
- **has_favorite_people** - Helps us make sure favorite people/contacts are configured correctly
- **has_group_calendar** - Helps us make sure if you have group calendars, they are configured correctly
- **has_group_calendar_account** - Helps us make sure if you have group calendars, they are configured correctly
- **has_group_toggled** - Helps us make sure if you have toggled group calendars, this setting is configured correctly
- **interesting_calendars_accounts_count** - Number of interesting calendar accounts you have to help us make sure your interesting calendar accounts are configured correctly
- **mail_accounts_count** - The total number of mail accounts in the sidebar to make sure this is configured correctly
- **mail_folder_type** - The type of folder the user tapped on to make sure it's configured correctly. This could include Deleted folder, Spam, or your Sent folder.

- **mail_inbox_unread_count** - Helps us ensure the unread count is displayed and configured accurately
- **mail_subfolder_depth** - Helps us ensure we can successfully display a user's mail subfolder configurations

StoreOp

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when Rights Management Service license store operation is performed.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.ApplicationScenarioId** - Scenario ID provided by the application
- **RMS.ContentId** - Content ID in End User License
- **RMS.Duration** - Total time for API call to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the operation
- **RMS.HttpCall** - indicate if there is HTTP operation
- **RMS.LicenseFormat** - The license Format: Xrml or Json
- **RMS.OperationName** - Operation name
- **RMS.Result** - Success or fail of the operation
- **RMS.ScenarioId** - Scenario ID defined by Rights Management Service Client
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.ServerType** - The type of Rights Management Service Server
- **RMS.StatusCode** - Status code of the operation result
- **RMS.Url** - The URL of Rights Management Service Server

Survey.Floodgate.TriggerMet

Tracks when a device has met the criteria to show a survey. Used to assess the health of the survey triggering process as well as to ensure the signal used to analyze customer issues and health is working properly.

The following fields are collected:

- **CampaignId** – Identifier of a service delivered campaign
- **SurveyId** – Unique instance of a campaign
- **SurveyType** – Identifies the type of survey

Survey.UI.Form.Submit

Tracks when a survey is submitted. Used to assess the health of the survey submission process as well as to

ensure the signal used to analyze customer issues and health is working properly.

The following fields are collected:

- **CampaignId** – Identifier of a service delivered campaign
- **SurveyId** – Unique instance of a campaign
- **SurveyType** – Identifies the type of survey

watchAppV2

This event is triggered from the Outlook watch app, when notifications are communicated from Outlook mobile to the Outlook watch app, and when the user is performing actions in the Outlook watch app. This event allows us to detect and fix possible issues with capabilities on the watch, such as receiving notifications and responding to emails.

The following fields are collected:

- **app_action** - Tells us the types of action the user took on the watch, such as "archive_message" to help us detect issues related to a specific action such as being unable to successfully archive messages on the watch
- **category** - Specifies a category (Usage, app_action, notification, etc.) for the event
- **is_complication_enabled** - Tells us if the user has added Outlook to their watch screen to help us detect issues related to watch screens
- **is_watch_app_installed** - Tells us if the user has installed our watch app on their device
- **notification** - Tells us the if a notification, if any was sent to the watch from the device.
- **view** - Tells us the view (Home, Inbox, Calendar, etc.) the watch was on, to help us detect issues related to a specific view
- **watch_app_version** - Tell us the version of the connected watch app
- **watch_manufacturer** - Tells us the manufacturer of the connected watch
- **watch_model** - Tells us the model of the connected watch
- **watch_os** - Tells us the OS version of the watch they have installed to help us detect issues related to specific OS versions of the watch
- **watch_os_brand** - Tells us the type of OS (Apple, Wear, Tizen, etc.) the connected watch is running

Application status and boot subtype

Determination if specific feature events have occurred, such as start or stop, and if feature is running.

app.startup

This event lets us detect and fix issues where Outlook is starting slowly or incompletely, making it difficult for users to use our app. This includes information on the specific features that were enabled and how long parts of the startup took.

The following fields are collected:

- **attach_base_context_millis** - time between the base Context starting and onCreate() starting
- **device_ram_in_mb** - the RAM available on the device
- **has_company_portal** - whether the company portal app is installed
- **hx_okhttp_mode** - whether the new email syncing service component is using OKHttp for sending and receiving HTTP-based network requests

- **initial_activity_name** - the Android Activity that launched the app
- **manufacturer** - the device manufacturer
- **model** - the device model
- **on_create_millis** - time taken in the onCreate() method
- **on_resume_millis** - time taken in the onResume() method
- **time_until_attach** - time between the class loading and the base Context starting
- **total_millis** - total time from class loading start to Android Activity resume completion

boot.time

This event lets us detect when critical app errors occurred that would cause your app to crash or experience serious issues like causing you to see empty rows in your inbox. This event collects information that allows us to categorize and classify issues to help prioritize the impact of issues on customers.

The following fields are collected:

- **black_list_reason** - Tells us if there is a reason why we should disregard this data. Some examples include launching due to a remote notification and launching due to a background fetch.
- **step_premain** – Tells us the amount of time it has taken for Outlook to go from the user tapping the icon to step0_main the "main" step defined in this document.
- **step0_main** - Tells us the amount of time it has taken for Outlook to get to the "main" step, which is a step defined by Apple.
- **step1_appWillFinishLaunching** - Tells us the amount of time it has taken for Outlook to go from the "main" step to the "appWillFinishLaunching" step, which is a step defined by Apple.
- **step2_appDidFinishLaunching** - Tells us the amount of time it has taken for Outlook to go from the "appWillFinishLaunching" step to the "appDidFinishLaunching" step, which is a step defined by Apple.
- **step3_engineStarted** - Tells us the amount of time it has taken for Outlook to go from the "appDidFinishLaunching" step to starting the engine of the app, which handles storing and syncing data.
- **step4_runLoopFirstIdle** - Tells us the amount of time it has taken for Outlook to go from the "engineStarted" step to having no additional work to complete.
- **total_time** - Tells us the total amount of time it has taken for Outlook to complete the boot process.

DnsLookupOp

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when DNS information lookup operation is performed.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.ApplicationScenarioId** - Scenario ID provided by the application

- **RMS.Duration** - Total time for the operation to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the operation
- **RMS.HttpCall** - indicate if there is http operation
- **RMS.LicenseFormat** - The license Format: Xrml or Json
- **RMS.NoOfDomainsSearched** - The number of domains searched
- **RMS.NoOfDomainsSkipped** - The number of domains skipped
- **RMS.Result** - Success or fail of the operation
- **RMS.ScenarioId** - Scenario ID defined by Rights Management Service Client
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.ServerType** - The type of Rights Management Service Server
- **RMS.StatusCode** - Status code of the operation result

first.visible

This event lets us detect the first time the app is launched intentionally by the user. This event is required to ensure that the app is successfully working in Original Equipment Manufacturer (OEM) builds.

The following fields are collected:

- **is_oem** - a field tracking that indicates whether the application is running on an OEM variant
- **is_system_install** - a field tracking the presence of a pre-installed property file that should indicate that this install is OEM
- **manufacturer** - device manufacturer
- **model** - device model
- **systemFlagSet** - value of the Android system flag (ApplicationInfo.FLAG_SYSTEM) that indicates if the application was installed as part of the device's system image

GetUserOp

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when getting user certificates operation is performed.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.ApplicationScenarioId** - Scenario ID provided by the application
- **RMS.ContentId** - Content ID
- **RMS.Duration** - Total time for the operation to complete

- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned from the operation
- **RMS.HttpCall** - indicate if there is HTTP operation
- **RMS.LicenseFormat** - The license Format: Xrml or Json
- **RMS.Result** - Success or fail of the operation
- **RMS.ScenarioId** - Scenario ID defined by Rights Management Service Client
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.ServerType** - The type of Rights Management Service Server
- **RMS.StatusCode** - Status code of the operation result
- **RMS.Type** - type of user info

HttpOp

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when http request operation is performed.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.ApplicationScenarioId** - Scenario ID provided by the application
- **RMS.CallBackStatus** - The status of authentication call back returned result
- **RMS.CallbackTime** - The time consumed by authentication call back
- **RMS.CorrelationId** - correlation ID of the http request
- **RMS.DataSize** - data size of the HTTP request
- **RMS.Duration** - Total time for the operation to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the operation
- **RMS.HttpCall** - indicate if there is nested http operation
- **RMS.LicenseFormat** - The license Format: Xrml or Json
- **RMS.OperationName** - operation name
- **RMS.Result** - Success or fail of the operation
- **RMS.ScenarioId** - Scenario ID defined by Rights Management Service Client
- **RMS.SDKVersion** - The version of Rights Management Service Client

- **RMS.ServerType** - The type of Rights Management Service Server
- **RMS.StatusCode** - Status code of the operation result
- **RMS.Url** - The URL of Rights Management Service Server
- **RMS.WinhttpCallbackStatus** - The status of winhttp call back result

Initialized

Allows us to analyze the health of the interface that allows mobile apps to fetch user and privacy settings from Office services and diagnose connectivity and privacy setting service issues.

The following fields are collected:

- **roamingSettingType** - identifies the location from which we attempt to read settings

IpcCreateOauth2Token

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when the IpcCreateOauth2Token API call is made.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.Duration** - Total time for API call to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the API call
- **RMS.HttpCall** - indicate if there is HTTP operation
- **RMS.Result** - Success or fail of the API call
- **RMS.ScenarioId** - Scenario ID defined by the API
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.StatusCode** - Status code of the returned result

Office.Android.AccountStorageInfo

This event determines the number of MSA and ADAL accounts in the registry and shared preferences. It enables the analysis of inconsistencies between data stores and helps us to stabilize app performance.

The following fields are collected:

- **RegistryADALCount**- Indicates number of ADAL accounts in registry.
- **RegistryLiveIdCount**- Indicates number of MSA accounts in registry.
- **SharedPrefADALCount**- Indicates number of ADAL accounts in shared preferences.
- **SharedPrefLiveIdCount**- Indicates number of MSA accounts in shared preferences.

Office.Android.AndroidOffice16BootLatency

Critical to capture for app performance metric with respect to the response time of the app from the boot. Microsoft uses this to collect the time taken for the app to be responsive and also detect scenarios that may impact boot time in Word, Excel, or PowerPoint.

The following fields are collected:

- **AppLaunchResponsiveTimeInMilliSec** - App launch responsive time
- **AppSuspendedDuringBoot** - Boolean to indicate if app was suspended during boot
- **CollectionTime** - Time of event
- **FileActivationAttempted** - Boolean to indicate if file activation as attempted
- **FirstIdleOnAppThreadTimeInMilliSec** - App thread idle time
- **IsThisFirstLaunch** - Boolean to indicate if this is first time app launch
- **UserDialogInterruptionDuringBoot** - Boolean to indicate if there was a blocking UI during boot

Office.Extensibility.OfficeJS.Appactivated

Records information about unexpected shutdowns of Office. This allows us to identify crashes or hangs in the product so that they can be addressed.

The following fields are collected:

- **Data_AirspaceInitTime:integer**- time taken to initialize Airspace Office component
- **Data_AllShapes:integer** - number of shapes in the document
- **Data_APIInitTime:integer** - time taken to initialize Visio API module
- **Data_AppSizeHeight** – Add-in window size's height
- **Data_AppSizeWidth** – Add-in window size's width
- **Data_AppURL** - URL of the Add-in; Logs full URL for Store Add-ins and URL domain for non-store Add-ins
- **Data_Doc_AsyncOpenKind:long** – Indicates whether a cached version of the cloud document was opened and which asynchronous refresh logic was used.
- **Data_AuthorsCount:integer** - number of authors who edited the document in this session
- **Data_BackgroundPages:integer** - number of background pages in diagram
- **Data_BootTime:integer** - The amount of time it took to boot Visio
- **Data_Browser** - Browser string with information about the browser such as type, version
- **Data_ChildWindowMixedModeTime:integer** - time taken to enable mixed mode in Visio (Child window can have different DpiAwareness from parent window)
- **Data_CommentsCount:integer** - number of comments in document
- **Data_ConnectionCount:integer** - number of data connections in diagram
- **Data_ContentMgrInitTim:integer** - time taken to initialize content manager
- **Data_CreateMainFrameTime:integer** - Create mainframe time
- **Data_CreatePaletteTime:integer** - Time taken to create the global color palette
- **Data_DispFormatCount:integer** - number of data graphics in diagram

- **Data_Doc_Ext:string** - Document extension
- **Data_Doc_Fqdn:string** - Where is document stored (SharePoint.com, live.net), only available for Office 365 domains
- **Data_Doc_FqdnHash:string** - Hash of where document is stored
- **Data_Doc_IsIncrementalOpen:bool**- : Was document opened incrementally (new feature that opens document without needing to download entire document)
- **Data_Doc_IsOpeningOfflineCopy:bool** - Is document being opened from local cache?
- **Data_Doc_IsSyncBacked:bool**- true when this is a server document that exists locally, and is synchronized with the server (for example, through OneDrive or ODB client apps)
- **Data_Doc_Location:long**- : Predefined set of values of where document is stored (Local, SharePoint, WOPI, Network etc.)
- **Data_Doc_LocationDetails:long** - Predefined set of values of more detailed location (Temp folder, downloads folder, One Drive Documents, One Drive Pictures
- **Data_Doc_ResourceIdHash:string** - Hash of resource Identifier for documents stored in cloud
- **Data_Doc_RtcType** - Indicates how the real-time channel (RTC) was setup for current file (Disabled, unsupported, on demand, always on, etc.).
- **Data_Doc_ServerDocId:string** - immutable identifier for documents stored in cloud
- **Data_Doc_SessionId:long** - generated GUID that Identifies the instance of the document within the same process session
- **Data_Doc_SizeInBytes:long** - Document size in bytes
- **Data_Doc_SpecialChars:long** - long Bitmask indicating special chars in the document's URL or Path
- **Data_Doc_SyncBackedType** - Indicator as to the type of document (local or service based)
- **Data_Doc_UrlHash:string** - hash of full URL of documents stored in cloud
- **Data_DpiAwarenessTime:integer** - Time taken to enable Per Monitor DPI Awareness
- **Data_DurationToCompleteInMilliseconds:double**- Duration to complete save as in millisecond
- **Data_ErrorCode:int** - 0 for success, integer for failure in save
- **Data_FailureReason:integer** - failure reason for asynchronous save
- **Data_FileExtension** - File extension of diagram opened
- **Data_FileHasDGMaster:bool** - true when file has Data Graphics
- **Data_FileHasImportedData:bool** - true when file has imported data
- **Data_FileHasShapesLinked:bool** - true when file has linked shapes to data
- **Data_FileIOBytesRead:int** - total bytes read while saving
- **Data_FileIOBytesReadSquared:int** - squared value of Data_FileIOBytesRead
- **Data_FileIOBytesWritten:int** - total bytes written while saving
- **Data_FileIOBytesWrittenSquared:int**- squared value of Data_FileIOBytesWritten
- **Data_FilePathHash:binary** -Binary Hash of file path

- **Data_FilePathHash: binary** - GUID for file path
- **Data_FileSize** - Document size in bytes
- **Data_ForegroundPages:integer** - number of foreground pages in diagram
- **Data_ForegroundShapes:integer** - integer number of shapes in Foreground pages
- **Data_GdiInitTime:integer** - time taken to initialize GDI module
- **Data_HasCoauthUserEdit:bool** - true if document was edited in a co-authoring session
- **Data_HasCustomPages:bool** - true if document contains custom pages
- **Data_HasCustPatterns:bool** - true if file has custom patterns
- **Data_HasDynConn:bool** - true if document contains dynamic connection
- **Data_HasScaledPages:bool** - true if document contains scaled pages
- **Data_HasUserWaitTime:bool** - true when file dialog is shown while saving
- **Data_InitAddinsTime:integer** - time taken to initialize and load the COM Add
- **Data_InitBrandTime:integer** - amount of time it takes to initialize splash screen and branding Office components
- **Data_InitGimmeTime:integer** - time taken to initialize Office component
- **Data_InitLicensingTime:integer** - time taken to initialize licensing Office component
- **Data_InitMsoUtilsTime:integer** - Initialization time to MSOUTILS Office component
- **Data_InitPerfTime:integer** - Performance Office component initialization time
- **Data_InitTCOTime:integer** - amount of time it takes to initialize Office component manager
- **Data_InitTrustCenterTime:integer** - Time taken to initialize Office component trust center
- **Data_InitVSSubSystemsTime:integer** - amount of time it takes to initialize Visio components
- **Data_InternalFile:bool** - true if file is an internal file. for example, Stencil
- **Data_IsAsyncSave:bool** - true if save was asynchronous
- **Data_IsAutoRecoveredFile:bool** - true if file was auto recovered
- **Data_IsEmbedded:bool** - true if Visio file was embedded in another app
- **Data_IsInfinitePageDisabledForAllPages:bool** - if Infinite Page disabled for all pages for the document true
- **Data_IsIRMProtected:bool** - true if file is Information Right Protected
- **Data_IsLocal:bool** - true if file is local
- **Data_IsRecoverySave:bool** - true if safe was triggered because of recovery
- **Data_IsShapeSearchPaneHiddenState:bool** - true if shape search pane was hidden for document
- **Data_IsSmartDiagramPresentInActivePageOfFile:bool** - bool, true if smart data visual diagram is present in active page of file
- **Data_IsSmartDiagramPresentInFile:bool** - bool, true if the smart data visual diagram present in file.
- **Data_IsUNC:bool** - true if document path is adhering to Universal Naming Convention

- **Data_LandscapePgCount:integer** - number of pages having landscape orientation in the diagram
- **Data_Layers:integer** - number of layers in the diagram
- **Data_LoadProfileTime:integer** - amount of time it takes to load Office profiler
- **Data_LoadRichEditTim:integer**- rich edit component load time
- **Data_LoadVisIntlTime:integer** - time taken to load Visio international DLL
- **Data_Location:integer** - Location of the file from which it was opened 0 Local, 1, Network, 2, SharePoint, 3 – Web
- **Data_MasterCount:integer** - number of masters in the diagram
- **Data_MaxCoauthUsers:integer** - maximum number of users co-authoring at any point of time in the session Filesystem, Registry, First Party, SDX
- **Data_MaxTilesAutoSizeOn:integer** - Maximum number of tiles of a page for which auto size was enabled
- **Data_MsoBeginBootTime:integer** - MSO boot time
- **Data_MsoDllLoadTime:integer** - time taken to load MSO DLL
- **Data_MsoEndBootTime:integer** - time taken to end MSO boot
- **Data_MsoInitCoreTime:integer** - Take taken to initialize MSO Office component
- **Data_MsoInitUITime:integer** - time taken to initialize MSO Office component UI
- **Data_MsoMigrateTime:integer** - Time taken to migrate user settings on first bootup if user upgraded from previous version
- **Data_NetworkIOBytesRead:int** - total network bytes read while saving
- **Data_NetworkIOBytesReadSquared:int** - squared value of Data_NetworkIOBytesRead
- **Data_NetworkIOBytesWritten:int** - total network bytes written while saving
- **Data_NetworkIOBytesWrittenSquared :int**- squared value of NetworkIOBytesWritten
- **Data_OartStartupTime:integer** - time taken to initialize OART Office component
- **Data_OleInitTime:integer** - OLE Office component initialization time
- **Data_OpenDurationTimeInMs:integer** - duration to open file in milliseconds
- **Data_OriginatedFromTemplateID:integer** - identifier for template from which diagram was created. NULL for third-party templates
- **Data_Pages:integer** - number of pages in document
- **Data_PositionToolbarsTime:integer** - time taken to get the toolbars into place
- **Data_ReadOnly:bool** - True if the file is read only
- **Data_RecordSetCount:integer** - number of record set in the diagram
- **Data_RecoveryTime:integer** - time taken to open recovery files
- **Data_ReviewerPages:integer** - number of reviewer pages in diagram
- **Data_RibbonTime:integer** - time taken to display the status bar

- **Data_RoamingSettingsStartupTime:integer** - time taken create and load all currently roamed Visio settings
- **Data_SchemeMgrStartupTime:integer** - time taken to initialize scheme manager
- **Data_SDX_AssetId** - ONLY exists for store Add-ins. OMEX gives the Add in an AssetId when it goes into Store
- **Data_SDX_BrowserToken** - Identifier that sits in the browser's cache
- **Data_SDX_HostJsVersion** - This is the platform-specific version of Office.js (for example, outlook web16.01.js) This contains the API surface for ad ins
- **Data_SDX_Id** - The GUID of an Add-in, which uniquely identifies it
- **Data_SDX_InstanceId** - Represents Add in document pair
- **Data_SDX_MarketplaceType** - Indicates where the Add-in installed from
- **Data_SDX_OfficeJsVersion** - This is the version of Office.js that will redirect to the platform-specific version.
- **Data_SDX_Version** - Version of the Add-in
- **Data_ShellCmdLineTime:integer** - time taken to Parse and execute any shell commands on the command line
- **Data_Size:long** - File size in bytes
- **Data_StartEndTransactionTime:integer** - time taken to initialize Visio components
- **Data_STNInitTime:integer** - time taken to initialize stencil window configuration
- **Data_Tag:string** - unique identifier to identify Save As event
- **Data_ThemeCount:integer** - number of themes in diagram
- **Data_TimeStamp** - Time stamp when document was closed
- **Data_UIInitTime:integer** - UI initialization time
- **Data_WasSuccessful:bool** - true if save as was successful
- **Data_WinLaunchTime:integer** - time taken to launch the Visio startup pane, etc.)

Office.Extensibility.Sandbox.ODPActivationHanging

Collects when an Office add-in takes unexpectedly long to launch (>5 sec). Used to detect and fix problems with Office add-ins launching.

The following fields are collected:

- **AppId** - ID of the App
- **AppInfo** - data regarding the type of add-in (task pane or UILess or in content etc.) and the provider type (omen, SharePoint, filesystem etc.)
- **AppInstanceId** - ID of the app instance
- **AssetId** - asset ID of the app
- **IsPreload** – indicates if the add-in is being preloaded in background for improving activation performance
- **NumberOfAddinsActivated** - counter of add-ins activated

- **RemoterType** - specifies the type of remoter (Trusted, untrusted, Win32webView, Trusted UDF etc.) used to activate the add-in
- **StoreType** - origin of the app
- **TimeForAuth** - time spent on auth
- **TimeForSandbox** - time spent on the sandbox
- **TimeForServerCall** - time spent on the server call
- **TotalTime** - total time spent
- **UsesSharedRuntime** - indicates if the app uses sharedRuntime or not.

Office.Lens.LensSdk.LaunchLens

Launches help us determine the number of users or devices launching the app and further understand feature usage. It helps us track the volume of users using the product, as well as identifying changes in trends, help look for and rectify issues in the product.

The following fields are collected on Android:

- **Data_CameraXBindUsecasesApi** - Time taken by camera library to initialize before it launches camera.
- **Data_CameraXBindUsecasesToPreview** - Time taken by camera library in showing camera preview to user.
- **Data_CurrentWorkFlowType** - Explains whether the user was capturing, editing, saving images, etc.
- **Data_IsDexModeEnabled** - Boolean indicating if device supports Samsung Dex features.
- **Data_IsEmbeddedLaunch** - Boolean field indicating if user launched the control in picture-in-picture mode.
- **Data_IsInterimCropEnabled** - Boolean field indicating if user has chosen to manually crop each image.
- **Data_IsMultiWindowEnabled** - Boolean field indicating if it's possible to run app in split screen.
- **Data_IsTalkBackEnabled** - Boolean indicating if device is in accessibility mode.
- **Data_LaunchPerf** - Integer indicating time taken to launch the app.
- **Data_LensSdkVersion** - Version of the SDK running in the app.
- **Data_RecoveryMode** - Boolean field indicating if this session was recovered after app was killed.
- **Data_SDKMode** - The mode in which image(s) were captured.
- **Data_SessionId** - Identifier tracking each session of the app.
- **Data_TelemetryEventTimestamp** - Time at which this event or action was completed.

The following fields are collected on iOS:

- **Data_currentWorkflowId** - Explains whether the user was capturing a photo, scanning document, whiteboard, etc; or extracting text, table, etc; from images.
- **Data_defaultWorkflow** - Explains the default mode in which camera was launched, like document, whiteboard, photo, businesscard.
- **Data_imageDPI** - Specifies the quality in DPI (low, high medium) in which media was captured.
- **Data_isExistingUser** - Specifies if the user is a new user or existing user.

- **Data_isFirstLaunch** - Boolean indicating if the app is being launched after a new installation.
- **Data_isResumeSession** - Specifies whether the app was launched in resume or user did a fresh start. (Boolean field)
- **Data_launchReason** - Determine if the launch is done via Camera or Gallery.
- **Data_launchWorkflowItem** - Field specifies if the app is launched from camera screen or edit screen.
- **Data_Lens_SessionId** - Identifier tracking each session of the app.
- **Data_LensEventName** - Name of the event, I.e. Office_Lens_LensSdk_LaunchLens
- **Data_mediaCompressionFactor** - The factor by which images are compressed by app.
- **Data_version** - Version of the SDK running in the app.

Office.OfficeMobile.AppActivation.Launch

This event identifies first time and subsequent activations through external triggers that activate the app. App activation loads certain dependencies that are responsible for making the app run smoothly and this event will record if it loaded successfully. It will also record the source of activation and app intent that was responsible for activating the app

The following fields are collected:

- **ActionName** - Integer value mapping to the name of the action/feature which is invoked from the activation point.
- **ActivationType** - Integer value mapping to the source of activation
- **IsActionTriggered** - Boolean value determining whether the action got triggered after the successful activation of the app.
- **IsFirstRun** - Boolean value determining whether it was the first run of the app or a subsequent run.

Office.OfficeMobile.FRE.FirstRunSetup

The first run of the app after installation will trigger this heartbeat event. It will help identify installs and auto upgrades from older versions of the app and enable us to identify errors in auto-upgrades, including library loads and expansion/language package download failures.

The following fields are collected:

- **IsFlightAssigned** - Boolean value determining if the user was part of any preassigned flight group which can trigger exposure to certain experiences.
- **IsFRELoadSuccessful** - integer mentioning the result state

Office.OneNote.Android.App.AppBootComplete, Office.Android.EarlyTelemetry.AppBootComplete

[This event was previously named OneNote.App.AppBootComplete.]

The critical signal used to ensure new consumer users (Microsoft Account) can successfully launch and use OneNote for the first time. This is used to ensure critical regression detection for OneNote app and service health. If users can't launch the app for the first time, this would trigger a high severity incident.

The following fields are collected:

- **ACTIVITY_BOOT_TIME_IN_MS** - Time taken to complete Activity creation
- **ACTIVITY_NAME** - Name of the Activity opened when booted
- **ANY_DIALOG_SHOWN** - Indicates if any dialog is shown during boot time
- **APP_SUSPEND_DURING_EVENT** - Indicates if the boot was preempted

- **APP_THREAD_CREATION_WAIT_TIME_TIME_FOR_APP_THREAD_CREATION** - Time taken to create Application threads
- **AVAILABLE_MEMORY_IN_MB** - Total memory available in device
- **AVG_SNAPSHOT_POPULATION_TIME** - Avg. time taken to fetch notebook structures while using the App
- **BOOT_END_AT_VIEW** - Sub-category of Activity name (Name of the View)
- **BOOT_SNAPSHOTS** - Detail of notebook structure fetches for the account(s) used in App
- **COREAPP_STARTUP_ACCOUNT_SETUP_STARTUP_ACCOUNT_SETUP** - Time taken to check and initiate SSO experience
- **CRASH_INTERACTION_DURING_BOOT> 0** - To indicate if the App crashed during last session
- **DALVIK_HEAP_LIMIT_IN_MB** - Obsolete
- **DELAY_LOAD_STICKY_NOTES** - Indicates if the Sticky Notes with delay or not
- **FISHBOWL_SHOWN_DURING_EVENT** - Indicates instances where content in not synced
- **HAS_LOGCAT_LOGGING_IMPACT_ON_BOOT** - Indicates if boot time is impacted due to logs
- **INIT_SNAPSHOT_DURATION** - Time taken to get the notebook structure for the user account(s)
- **IS_COLD_BOOT** - Indicates if the App launch when the App was not running in the background
- **IS_FIRST_LAUNCH** - Indicates if this is the First time App was launched in the device
- **IS_FOLDABLE_TYPE** - Indicates if the device is a foldable device
- **IS_PHONE** - Indicates if the device is a phone or tablet
- **IS_RECENT_PAGES_AVAILABLE_ON_FRAGMENT_CREATION** - Indicates if UI is ready and waiting for content to be made available
- **IS_REHYDRATE_LAUNCH** - Indicates if App was killed by the system
- **IS_UPGRADE** - Indicates if the App is being launched after upgrade
- **JOT_MAIN_APP_CREATE_TIME_MAIN_APP_CREATE_TIME** - Time taken to create JOT component (shared code component)
- **JOT_MAIN_APP_INIT_TIME_MAIN_APP_INIT_TIME** - Time taken to initialize JOT component
- **LAUNCH_POINT** - Indicates if the App is opened from Widget or App Icon or Hyperlink or Share to ON etc.
- **MSO_ACTIVATION_TIME_ACTIVATION_TIME** - Time taken to initialize MSO
- **NATIVE_LIBRARIES_LOAD_TIME** - Time taken to load libraries
- **NAVIGATION_CREATE_TO_NAVIGATION_RESUME_CREATE_TO_NAVIGATION_RESUME** - Time taken to complete navigation
- **NAVIGATION_RESUME_TO_BOOT_END_RESUME_TO_BOOT_END** - Time taken to measure delay in page load after boot
- **NAVIGATION_SET_CONTENT_VIEW_TIME_SET_CONTENT_VIEW_TIME** - Time taken to bring in content
- **NUMBER_Of_RUNNING_PROCESSES** - Indicates number of active processes running

- **NUMBER_OF_SNAPSHOTS** - Count of fetches of notebook structure during boot
- **OFFICEASSETMANAGER_INITIALIZATION_TIME** - Time taken to unzip and initialize Asset Manager
- **PROCESS_BOOT_TIME_IN_MS** - Time taken to complete Process creation
- **ROOT_ACTIVITY_CREATE_ACTIVITY_CREATE** - Time taken to transition from Root layer
- **ROOT_ACTIVITY_DISK_CHECK_ACTIVITY_DISK_CHECK** - Obsolete
- **ROOT_ACTIVITY_LAUNCH_NEXTACTIVITY_ACTIVITY_LAUNCH_NEXTACTIVITY** - Obsolete
- **ROOT_ACTIVITY_PROCESS_INTENT_ACTIVITY_PROCESS_INTENT** - Obsolete
- **ROOT_ACTIVITY_SESSION_ACTIVITY_SESSION** - Time taken to transition from Root layer
- **ROOT_TO_NAVIGATION_TRANSITION_TO_NAVIGATION_TRANSITION** - Time taken to handle navigation from Root
- **SNAPSHOT_PUBLISH_TO_RENDERING_END_PUBLISH_TO_RENDERING_END** - Time to complete rendering content
- **SPLASH_ACTIVITY_SESSION_ACTIVITY_SESSION** - Time taken to show splash screen
- **SPLASH_TO_ROOT_TRANSITION_TO_ROOT_TRANSITION** - Time taken to transition from Root layer
- **TIME_BETWEEN_PROCESS_BOOT_AND_ACTIVITY_BEGIN_IN_MS** - Time between process & activity creation
- **TIME_TAKEN_IN_MS** - Time taken to complete boot
- **TOTAL_MEMORY_IN_MB** - Total memory of the Device
- **USER_INTERACTED_DURING_EVENT** - Indicates if the user has interacted during booting

Office.OneNote.Android.App.OneNoteAppForeground, Office.Android.EarlyTelemetry.OneNoteAppForeground

[This event was previously named OneNote.App.OneNoteAppForeground.]

The signal used to indicate OneNote App is in foreground. The telemetry is used to ensure critical regression detection for OneNote app and service health.

The following fields are collected:

- None

Office.OneNote.Android.AppLaunch, Office.Android.EarlyTelemetry.AppLaunch

[This event was previously named OneNote.AppLaunch.]

The critical signal used to ensure OneNote users can successfully launch the app. The telemetry is used to ensure critical regression detection for OneNote app and service health.

The following fields are collected:

- **ANDROID_SDK_VERSION** - Captures the Android SDK Version
- **FirstLaunchTime** - Records time when the App was first launched
- **InstallLocation** - Indicates if the app is pre-installed or downloaded form Store
- **is_boot_completed_ever** - Indicates if the App has even been booted successfully before in the device
- **IS_DARK_MODE_ENABLED** - A Boolean which indicates if app is in dark mode or not

- **NewOneNoteUser** - Identify if the user is a new user

Office.Outlook.Desktop.ExchangePuidAndTenantCorrelation

Collects the user PUID and Tenant Identifier once per session. The correlation of PUID and Tenant are necessary to understand and diagnose Outlook issues on a per-tenant basis.

The following fields are collected:

- **CollectionTime** - Timestamp of the event
- **ConnId** - Connection Identifier: Identifier of the connection parsing out PUID and OMS tenant Identifier
- **OMSTenantId** – Microsoft-generated Unique identifier of Tenant
- **PUID** - Exchange's PUID to uniquely identify users

Office.Outlook.Mac.MacOLKActivationState

Collects how Outlook is activated, such as with a subscription or volume license. The data is monitored to ensure we don't see any spikes in failures. We also analyze the data to find areas of improvement.

The following fields are collected:

- **SetupUIActivationMethod** - Outlook activation method, such as subscription or volume license

Office.PowerPoint.DocOperation.Open

Collected whenever PowerPoint opens a file. Contains success information, failure details, performance metrics, and basic details about the file including file format type and document metadata. This information is necessary to ensure PowerPoint can open files successfully with no degradation in performance. It allows us to diagnose any problems we discover.

The following fields are collected:

- **Data_AddDocTelemetryResult** - Does this log entry have all necessary document telemetry (Data_Doc_* fields)
- **Data_AddDocumentToMruList** - Method AddDocumentToMruList execution duration
- **Data_AlreadyOpened** - Was this document previously opened (within the context of the same process session)
- **Data_AntiVirusScanMethod** - Predefined set of values of type of anti-virus scanned (IOAV, AMSI, None etc.)
- **Data_AntiVirusScanStatus** - Predefined set of values of anti-virus scan that happens for every document opened (NoThreatsDetected, Failed, MalwareDetected, etc.)
- **Data_AsyncOpenKind** - Predefined set of values of async options (Collab, ServerOnly, SyncBacked, NotAsync)
- **Data_CancelBackgroundDownloadHr** - Was downloading of document interrupted? If yes, what was the result of interruption?
- **Data_CheckForAssistedReadingReasons** - Method CheckForAssistedReadingReasons execution duration in milliseconds
- **Data_CheckForDisabledDocument** - Method CheckForDisabledDocument execution duration in milliseconds
- **Data_CheckForExistingDocument** - Method CheckForExistingDocument execution duration in milliseconds
- **Data_CheckIncOpenResult** - Method CheckIncOpenResult execution duration in milliseconds

- **Data_CheckLambdaResult** - Method CheckLambdaResult execution duration in milliseconds
- **Data_CheckPackageForRequiredParts** - Method CheckPackageForRequiredParts execution duration in milliseconds
- **Data_CheckPackageForSpecialCases** - Method CheckPackageForSpecialCases execution duration in milliseconds
- **Data_CheckRequiredPartsLoaded** - Method CheckRequiredPartsLoaded execution duration in milliseconds
- **Data_CheckWebSharingViolationForIncOpen** - Method CheckWebSharingViolationForIncOpen execution duration in milliseconds
- **Data_CloseAndReopenWithoutDiscard** – Whether a document was closed and reopened during the open process without discarding.
- **Data_ClpDocHasDrmDoc:bool** - Whether the document has a DRM document
- **Data_ClpDocHasIdentity:bool** - Whether the document has identity info (used to get and set sensitivity labels)
- **Data_ClpDocHasSessionMetadata:bool** – Whether the document has working sensitivity label metadata from the session
- **Data_ClpDocHasSpoMetadata:bool** - Whether the document has sensitivity label metadata from SPO via IMetadataCache
- **Data_ClpDocHasSpoPackage:bool** – Whether the document has sensitivity label metadata from SPO via IPackage
- **Data_ClpDocIsProtected:bool** - Whether or not the document is protected by IRM
- **Data_ClpDocMetadataSource:int** - Enum specifying where sensitivity label metadata is from (IRM, OPC part, Sharepoint etc)
- **Data_ClpDocNeedsUpconversion:bool** - Whether the document needs to upconvert sensitivity label data from the custom.xml part
- **Data_ClpDocNumFailedSetLabels:int** - Count of sensitivity labels that failed to set on the document
- **Data_ClpDocSessionMetadataDirty:bool** - Whether the document has working sensitivity label metadata that has been dirtied
- **Data_ClpDocWasInTrustBoundary:bool** – Whether the document was in the trust boundary (allowing for coauthoring on documents protected by sensitivity labels)
- **Data_ContentTransaction** - Predefined set of values of when transaction can be created (AllowedOnLoadDocument, AllowedOnOpenComplete, etc.)
- **Data_CorrelationId** - A GUID passed to PowerPoint by ProtocolHandler for correlating telemetry. ProtocolHandler is a separate process that handles Office links for the OS.
- **Data_CppUncaughtExceptionCount:long** - Uncaught native exceptions while activity was running
- **Data_CreateDocumentTimeMS** - Method CreateDocumentTimeMS execution duration in milliseconds
- **Data_CreateDocumentToken** - Method CreateDocumentToken execution duration in milliseconds
- **Data_CreateDocumentToW** - Method CreateDocumentToW execution duration in milliseconds
- **Data_CreateDocWindow** - Method CreateDocWindow execution duration in milliseconds

- **Data_CreateLocalTempFile** - Method CreateLocalTempFile execution duration in milliseconds
- **Data_CsiDownloadErrDlgSuppressed:bool** – Whether the dialog that would have been shown by CSI during a download error has been suppressed, usually in favor of a dialog shown by PowerPoint instead.
- **Data_DetachedDuration:long** - Time for which Activity was detached/not running
- **Data_DetermineFileType** - Method DetermineFileType execution duration in milliseconds
- **Data_Doc_AccessMode:long** - How was this document opened (Read only / read write)
- **Data_Doc_AssistedReadingReasons:long** - Predefined set of values of why document was opened in assisted reading mode
- **Data_Doc_AsyncOpenKind:long** – Indicates whether a cached version of the cloud document was opened and which asynchronous refresh logic was used.
- **Data_Doc_ChunkingType:long** - How is document stored in SharePoint
- **Data_Doc_EdpState:long** - Enterprise Data Protection state of document
- **Data_Doc_Ext:string** - Document extension
- **Data_Doc_Extension:string** - Document extension
- **Data_Doc_FileFormat:long** - Predefined set of values of format of file (more granular than extension)
- **Data_Doc_Fqdn:string** – - Where is document stored (SharePoint.com, live.net), only available for Office 365 domains
- **Data_Doc_FqdnHash:string** – - Hash of where document is stored
- **Data_Doc_IdentityTelemetryId:string** – - Unique GUID of user
- **Data_Doc_IdentityUniqueId:string** - Unique identifier of identity that was used for Shared Documents action
- **Data_Doc_IOFlags:long** - Bitmask for various IO-related flags for a given document
- **Data_Doc_IrmRights:long** - Predefined set of values of what type of Information Rights Management is applied on this document (Forward, Reply, SecureReader, Edit etc.)
- **Data_Doc_IsCloudCollabEnabled:bool** - True if the "IsCloudCollabEnabled" HTTP header has already been received from an OPTIONS request.
- **Data_Doc_IsIncrementalOpen:bool** – - Was document opened incrementally (new feature that opens document without needing to download entire document)
- **Data_Doc_IsOcsSupported:bool** - Is Document supports co-authoring using new OCS service
- **Data_Doc_IsOpeningOfflineCopy:bool** - Is document being opened from local cache?
- **Data_Doc_IsSyncBacked:bool** - Is document opened from folder that is using OneDrive sync back app
- **Data_Doc_Location:long** - Predefined set of values of where document is stored (Local, SharePoint, WOPI, Network etc.)
- **Data_Doc_LocationDetails:long** - Predefined set of values of more detailed location (Temp folder, downloads folder, One Drive Documents, One Drive Pictures, etc.)
- **Data_Doc_NumberCoAuthors:long** - Number of co-authors at the time of opening of a document
- **Data_Doc_PasswordFlags:long** - Predefined set of values of how document is encrypted with

password (None, password to read, password to edit)

- **Data_Doc_ReadOnlyReasons:long** –- Predefined set of values of why this document was marked read only (Locked on server, final document, password protected to edit, etc.)
- **Data_Doc_ResourceIdHash:string** - Hash of resource identifier for documents stored in cloud
- **Data_Doc_RtcType** - Indicates how the real-time channel (RTC) was set up for current file (Disabled, unsupported, on demand, always on, etc.).
- **Data_Doc_ServerDocId:string** - immutable identifier for documents stored in cloud
- **Data_Doc_ServerProtocol:long** - Predefined set of values of which protocol is used to talk to server (Http, Cobalt, WOPI etc.)
- **Data_Doc_ServerType:long** - Predefined set of values of type of server (SharePoint, DropBox, WOPI)
- **Data_Doc_ServerVersion:long** - Is server is based off Office14, Office15, Office 16?
- **Data_Doc_SessionId:long** - generated GUID that Identifies the instance of the document within the same process session
- **Data_Doc_SharePointServiceContext:string** - An opaque string, typically GridManagerID.FarmID. Useful for correlating client-side and server-side logs
- **Data_Doc_SizeInBytes:long** - Document size in bytes
- **Data_Doc_SpecialChars:long** - Bitmask indicating special chars in the document's URL or Path
- **Data_Doc_StorageProviderId:string** - A string that identifies the document's storage provider, like "DropBox"
- **Data_Doc_StreamAvailability:long**- Predefined set of values of status of document Stream (available, permanently disabled, not available)
- **Data_Doc_UrlHash:string** - hash of full URL of documents stored in cloud
- **Data_Doc_UsedWrsDataOnOpen:bool** - true if the file was opened incrementally using pre-cached WRS data on the host
- **Data_Doc_WopiServiceId:string** - WOPI Service identifier, for example "Dropbox"
- **Data_DownloadErrorCsi:int** – Type of a download error, as provided by CSI
- **Data_DownloadErrorHResult:int** – HResult of a download error, as provided by CSI
- **Data_DownloadExcludedData** - Method DownloadExcludedData execution duration in milliseconds
- **Data_DownloadExcludedDataTelemetry** - Predefined set of values of state of synchronously waiting for download(SynchronousLogicHit, UserCancelled RunModalTaskUnexpectedHResult etc.)
- **Data_DownloadFileInBGThread** - Method DownloadFileInBGThread execution duration in milliseconds
- **Data_DownloadFragmentSize** - Size of fragment(downloadable chunk of file), usually 3.5 MB
- **Data_ExcludedEmbeddedItems** - Number of zip parts that are excluded for first render
- **Data_ExcludedEmbeddedItemsSize** - Total size of zip parts that are excluded for first render
- **Data_ExcludedRequiredItems** - Number of zip parts that are excluded but required for first render
- **Data_ExcludedRequiredItemsSize** - Total size of zip parts that are excluded but required for first

render

- **Data_ExecutionCount** - How many times IncOpen protocol was executed
- **Data_FailureComponent:long** - Predefined set of values of which component caused this protocol to fail? (Conflict, CSI, Internal etc.)
- **Data_FailureReason:long** - Predefined set of values of what's the failure reason (FileIsCorrupt, BlockedByAntivirus etc.)
- **Data_FCreateNew** - Is this new blank document
- **Data_FCreateNewFromTemplate** - Is this new document from templates
- **Data_FErrorAfterDocWinCreation:boolean**- Did any error or exception happen after the document window is created.
- **Data_FileIOClpState:int** – Bitset containing values regarding sensitivity label status. For example, this includes information about whether coauthoring with protected labels is enabled, whether the document has a label applied from the current tenant, and whether the document is protected by IRM.
- **Data_FileUrlLocation** - Predefined set of values of where document is stored (NetworkShare, LocalDrive, ServerOther etc.)
- **Data_FirstSlideCompressedSize** - compressed size of first slide zip part (usually Slide1.xml)
- **Data_FIsAutoBackupFile**- Is this file an auto backup file?
- **Data_FIsDownloadFileInBgThreadEnabled** - Is downloading in background thread enabled?
- **Data_fLifeguarded:bool** - Was document ever lifeguarded (feature to fix document errors by themselves without prompting user)?
- **Data_ForceReopenOnIncOpenMergeFailure** - Flag representing if we were forced to reopen due to merge failure in Inc Open
- **Data_ForegroundThreadPass0TimeMS** - (Mac only) Total time spent in foreground thread in first pass
- **Data_ForegroundThreadPass1TimeMS** - (Mac only) Total time spent in foreground thread in second pass
- **Data_FWebCreatorDoc** - Is doc created from template or QuickStarter
- **Data_HasDocToken** - Does this document have CSI doc token (internal code)
- **Data_HasDocument** - Does this document have CSI document (internal code)
- **Data_InclusiveMeasurements** - Does method measurement durations are inclusive of child method call duration
- **Data_IncompleteDocReasons** - Predefined set of values of why open was incomplete (Unknown, DiscardFailure etc.)
- **Data_IncOpenDisabledReasons** - Predefined set of values of Reasons why incremental open was disabled
- **Data_IncOpenFailureHr** - result of why Incremental open failed
- **Data_IncOpenFailureTag** - Tag (pointer to code location) of where Incremental open failed
- **Data_IncOpenFallbackReason** - Why was IncOpen not run

- **Data_IncOpenRequiredTypes** - Predefined set of values of content types needed for first render (RequiredXmlZipItem, RequiredNotesMaster etc.)
- **Data_IncOpenStatus** - Predefined set of values of Incremental open status (Attempted, FoundExcludedItems, DocIncOpenInfoCreated etc.)
- **Data_InitFileContents** - Method InitFileContents execution duration in milliseconds
- **Data_InitialExcludedItems** - Number of zip parts that are excluded initially
- **Data_InitialExcludedItemsSize** - Total size of zip parts that are excluded initially
- **Data_InitializationTimeMS** - (Mac Only) Time to initialize
- **Data_InitialRoundtripCount** - Number of server responses needed to form initial zip archive
- **Data_InitLoadProcess** - Method InitLoadProcess execution duration in milliseconds
- **Data_InitPackage** - Method InitPackage execution duration in milliseconds
- **Data_InitSecureReaderReasons** - Method InitSecureReaderReasons execution duration in milliseconds
- **Data_IsIncOpenInProgressWhileOpen** - In case of multiple open of the same document, is Inc open protocol running alongside open protocol?
- **Data_IsMultiOpen** - Do we support multiple open?
- **Data_IsOCS** - Was document in OCS mode in its' last known state
- **Data_IsODPFile** - Is document in 'Open Document Format' used by OpenOffice.org
- **Data_IsPPTMetroFile** - Is document metro (pptx) file format
- **Data_LoadDocument** - Method LoadDocument execution duration in milliseconds
- **Data_MeasurementBreakdown** - Encoded measurement breakdown (shortened detailed method perf)
- **Data_Measurements** - Encoded measurements
- **Data_MethodId** - Last method that was executed
- **Data_NotRequiredExcludedItems** - Total number of PowerPoint package items that are not required for first render and excluded
- **Data_NotRequiredExcludedItemsSize** - Total size of PowerPoint package items that are not required for first render and excluded
- **Data_NotRequiredExcludedParts** - Total number of zip parts that are not required for first render and excluded
- **Data_NotRequiredExcludedPartsSize** - Total number of zip parts that are not required for first render and excluded
- **Data_OngoingBlockingOpenCount** – This is a count of how many blocking open protocols are currently running.
- **Data_OngoingOpenCount** – This is a count of how many open protocols are currently running.
- **Data_OpenCompleteFailureHR** - result of why OpenComplete protocol failed
- **Data_OpenCompleteFailureTag** - Tag (pointer to code location) of where OpenComplete protocol

failed

- **Data_OpenLifeguardOption** - Predefined set of values of choices for lifeguard operation (None, TryAgain, OpenInWebApp etc.)
- **Data_OpenReason** - Predefined set of values of how this document was opened (FilePicker, OpenFromMru, FileDrop etc.)
- **Data_OSRPolicy** - SecureReader Policy
- **Data_OSRReason** - Reasons why this document was opened in Secure Reader
- **Data_OtherContentTypesWithRequiredParts** - Nonstandard content types that were excluded but required for first render
- **Data_PrepCacheAsync** - Flag for OcsiOpenPerfPrepCacheAsync
- **Data_PreviousDiscardFailed** - Indicates previous open/close attempt on the document didn't properly release all memory
- **Data_PreviousFailureHr** - In case of reopening of the same document, what was last failure result
- **Data_PreviousFailureTag** - In case of reopening of the same document, what was last failure tag (pointer to code location)
- **Data_RemoteDocToken** - Is Remote Open enabled (prototype feature that enables opening file from service rather than from host)?
- **Data_Repair** - Are we in document repair mode (for corrupt documents that are fixable)
- **Data_RequestPauseStats** - How many times code requested to pause perf recording
- **Data_RequiredPartsComressedSize** - Total size of required PowerPoint parts needed for first render
- **Data_RequiredPartsDownload** - Total size of required PowerPoint parts that are downloaded
- **Data_RequiredPartsRoundtripCount** - Total number of roundtrips (calls to host) needed to get all the required PowerPoint parts for first render
- **Data_RequiredZipItemsDownload** - Total size of required zip items needed for first render
- **Data_RequiredZipItemsRoundtripCount** - Total number of roundtrips (calls to host) needed to get all the required zip items for first render
- **Data_RoundtripsAfterMissingRequiredParts** - Total number of roundtrips (calls to host) needed after we found missing required PowerPoint parts
- **Data_RoundtripsAfterMissingRequiredZipItems** - Total number of roundtrips (calls to host) needed after we found missing required zip items
- **Data_RoundtripsAfterRequiredPackage** - Total number of roundtrips (calls to host) needed after we created the package
- **Data_RoundtripsAfterRequiredParts** - Total number of roundtrips (calls to host) needed after we downloaded all required parts
- **Data_SetDocCoAuthAutoSaveable** - Method SetDocCoAuthAutoSaveable execution duration in milliseconds
- **Data_SniffedFileType** - An educated guess of proposed file type of corrupt document
- **Data_StartTime** - Perf counter when Open started

- **Data_StopwatchDuration:long** - Total time for Activity
- **Data_SyncSlides** - Method SyncSlides execution duration in milliseconds
- **Data_TimerStartReason** - Predefined set of values of how timer was started (CatchMissedSyncStateNotification, WaitingForFirstDownload etc.)
- **Data_TimeSplitMeasurements** - Encoded measurement breakdown (shortened detailed method perf)
- **Data_TimeToInitialPackage** - Time took to create initial package
- **Data_TimeToRequiredPackage** - Time took to create required package
- **Data_TimeToRequiredParts** - Time took to create package with all required parts in it
- **Data_TotalRequiredParts** - Total number of PowerPoint parts required for first render
- **Data_UnknownRequiredParts** - Total number of non-standard parts required for first render
- **Data_UnpackLinkWatsonId** - Watson identifier of error when document is opened via Share OneDrive URL
- **Data_UnpackResultHint** - Predefined set of values of unpacking share URL results (NavigateToWebWithoutError, UrlUnsupported, AttemptOpen etc.)
- **Data_UnpackUrl** - Method UnpackUrl execution duration in milliseconds
- **Data_UpdateAppstateTimeMS** - Method UpdateAppstate execution duration in milliseconds
- **Data_UpdateCoauthoringState** - Method UpdateCoauthoringState execution duration in milliseconds
- **Data_UpdateReadOnlyState** - Method UpdateReadOnlyState execution duration in milliseconds
- **Data_WACCorrelationId** - In case of opening file in browser, get the correlation of WebApp logs
- **Data_WasCachedOnInitialize** - Was this document cached during initialization
- **Data_WBDirtyBeforeDiscard** - Is working branch became dirty before reopening document
- **Data_ZRTOpenDisabledReasons** - Why we could not open document from cache (Zero Round Trip)

Office.PowerPoint.PPT.Desktop.Bootime

Collecting how PowerPoint is booted. It includes boot PowerPoint in protected view, in assisted reading mode, from Macro, print, new and blank document, document recovery, from automation and if it is click- to-run. It also collects how long it takes PowerPoint to boot. This data is critical to guarantee PowerPoint performs well when booted from different modes. Microsoft uses this data to catch long booting time when opening PowerPoint from different modes.

The following fields are collected:

- **AssistedReading** - in assisted reading mode
- **Automation** - from automation
- **Benchmark** - run performance benchmark
- **Blank** - blank document
- **BootTime** - session boot time
- **Embedding** - embedding document
- **IsC2R** - is click-to-run

- **IsNew** - new document
- **IsOpen** - is open
- **Macro1** - run Macro
- **Macro2** - run Macro
- **NonStandardSpaceInCmdLine** – there is non-standard space in command line
- **Print** - print document
- **PrintDialog** - print document with dialog
- **PrintPrinter** - print document with printer
- **ProtectedView** - in protected view
- **Regserver** - Register PowerPoint as a COM server
- **Restore** - restore document
- **Show** - show document
- **Time** - session time
- **UnprotectedView** - in unprotected view

Office.PowerPoint.PPT.HasUserEditedDocument

Collected when a user starts editing a document. Microsoft uses this data to calculate active users who edited a PowerPoint document

The following fields are collected:

- **CorrelationId** – document Correlation Identifier

Office.Project.BootAndOpenProject

Project is booted by opening a file. This event indicates that a user has opened Office Project with an associated file. It contains critical success data of making sure Project can start and load a file.

The following fields are collected:

- **Data_AlertTime** - The amount of time the boot dialog was active.
- **Data_BootTime** - The amount of time it took to boot Project
- **Data_CacheFileSize** - If the file was cached, the size of the file
- **Data_EntDocType** - The type of file that was opened
- **Data_IsInCache** - Whether the file opened was cached
- **Data_LoadSRAs** - If the user wants to load SRAs or not
- **Data_Outcome** - Total boot and file open time.
- **Data_OpenFromDocLib** - If the Project file opened was from the document library
- **Data_ProjectServerVersion** - The version and build that Project is currently on

Office.Project.BootProject

Project is booted without opening a file. This event indicates that a user has opened Office Project without an associated file. It contains critical success data of making sure Project can start.

The following fields are collected:

- **Data_BootTime** - The amount of time it took to boot Project
- **Data_FileLoaded** - False if opening from out-space or new blank project
- **Data_IsEntOfflineWithProfile** - If the users are in the professional SKU and not connected to the server
- **Data_IsEntOnline** - If the session of Project is connected to a Project server with enterprise features
- **Data_IsLocalProfile** - If the Project session is connected to a Project server with enterprise features
- **Data_ProjectServerVersion** - The version and build that Project is currently on

Office.Project.OpenProject

Project opens a file. This event indicates a user directly opening a Project file by a user. It contains critical success data of opening files in Project.

The following fields are collected:

- **Data_AgileMode** - defines if the project opened is a waterfall or agile project
- **Data_AlertTime** - The amount of time the boot dialog was active
- **Data_CacheFileSize** - If the file was cached, the size of the file
- **Data_EntDocType** - the type of file that was opened
- **Data_IsInCache** - whether the file opened was cached
- **Data_LoadSRAs** - If the user wants to load SRAs or not
- **Data_OpenFromDocLib** - If the Project file opened was from the document library
- **Data_Outcome** - Total boot and file open time
- **Data_Outcome** - Total boot and file open time.
- **Data_ProjectServerVersion** - The version and build that Project is currently on

Office.SessionIdProvider.OfficeProcessSessionStart

Applicable to all the Office windows-based applications: win32 and UWP

The following fields are collected:

- **OfficeProcessSessionStart** sends basic information upon the start of a new Office session. This is used to count the number of unique sessions seen on a given device. This is used as a heartbeat event to ensure that the application is running on a device or not. In addition, it serves as a critical signal for overall application reliability
- **AppSessionGuid** - An identifier of a particular application session starting at process creation time and persisting until process end. It is formatted as a standard 128-bit GUID but constructed of four parts. Those four parts in order are (1) 32-bit Process ID (2) 16-bit Session ID (3) 16 bit Boot ID (4) 64-bit Process creation time in UTC 100 ns
- **processSessionId** - Randomly generated guid to identify the app session
- **UTCReplace_AppSessionGuid** - Constant boolean value. Always true.

Office.System.SessionHandoff

Indicates that the current Office session is a handoff session. This means that handling of a user's request to open a document is being handed off to an already running instance of the same application.

The following fields are collected.

- **ParentSessionId** - The ID of the session that will be taking over handling of the users' request.

Office.TelemetryEngine.IsPreLaunch

Applicable for Office UWP applications. This event is fired when an Office application is initiated for the first-time post upgrade/install from the store. This is part of basic diagnostic data, used to track whether a session is launch session or not.

The following fields are collected:

- **appVersionBuild** - The app build version number.
- **appVersionMajor** - The app major version number.
- **appVersionMinor** - The app minor version number.
- **appVersionRev** - The app revision version number.
- **SessionID** - Randomly generated GUID to identify the app session

Office.TelemetryEngine.SessionHandOff

Applicable to Win32 Office applications. This event helps us understand whether there was a new session created to handle a user-initiated file open event. It is a critical diagnostic information that is used to derive reliability signal and ensure that the application is working as expected.

The following fields are collected:

- **appVersionBuild** - The app build version number.
- **appVersionMajor** - The app major version number.
- **appVersionMinor** - The app minor version number.
- **appVersionRev** - The app revision version number.
- **childSessionID** - Randomly generated guid to identify the app session
- **parentSessionId** - Randomly generated guid to identify the app session

Office.Visio.VisioIosAppBootTime

This is triggered every time the Visio iOS application boots. It is essential to understand boot performance of the Visio iOS app. Used to troubleshoot poor performance.

The following fields are collected:

- **Data_AppBootTime** - Duration taken for app to boot, in milliseconds.

Office.Visio.VisioIosAppResumeTime

This event is triggered every time the Visio iOS app resumes focus. It is essential to measure app resume performance and troubleshoot performance issues of Visio iOS app.

The following fields are collected:

- **Data_AppResumeTime** - Duration for app to resume in milliseconds.

Office.Word.FileOpen.OpenCmdFileMruPriv

This event indicates Office Word opens a document from the Most Recent Used (MRU) list. It also contains critical file open performance data and is an app start event from user perspective. The event monitors whether fileopen-from-MRU is working as expected. It is also used to calculated monthly active users/devices, and cloud reliability metrics.

The following fields are collected:

- **Data_AddDocTelemRes** - Reports whether we were able to properly populate other document

telemetry-related values in the event. Used for data quality diagnostics.

- **Data_BytesAsynchronous** - Number of bytes (compressed) that we believe we can open the file without if we get them before the user wants to start editing or maybe saving
- **Data_BytesAsynchronousWithWork** - Number of bytes (compressed) that we might be able to open the file without but would require significant code investments to make it happen
- **Data_BytesSynchronous** - Number of bytes (compressed) that we must have before we can start opening the file
- **Data_BytesUnknown** - Number of bytes in document parts that we don't expect to find
- **Data_DetachedDuration** - How long was the activity detached from the thread
- **Data_Doc_AccessMode** - Document is read only/editable
- **Data_Doc_AssistedReadingReasons** - Predefined set of values of why document was opened in assisted reading mode
- **Data_Doc_AsyncOpenKind** – Indicates whether a cached version of the cloud document was opened and which asynchronous refresh logic was used.
- **Data_Doc_ChunkingType** - Units used for incremental document open
- **Data_Doc_EdpState** - Electronic Data Protection setting for the document
- **Data_Doc_Ext** - Document extension (docx/xlsb/pptx etc.)
- **Data_Doc_FileFormat** - File format protocol version
- **Data_Doc_Fqdn** - OneDrive or SharePoint Online Domain Name
- **Data_Doc_FqdnHash** - One-way hash of customer identifiable domain name
- **Data_Doc_IdentityTelemetryId** - A one-way hash of the user identity used to perform the open
- **Data_Doc_InitializationScenario** - Records how the document was opened
- **Data_Doc_IOFlags** - Reports on the cached flags used to set open request options
- **Data_Doc_IrmRights** - Actions permitted by the Electronic Data Protection policy that has been applied to the document/user
- **Data_Doc_IsIncrementalOpen** - Flag indicating that the document has been incrementally opened
- **Data_Doc_IsOcsSupported** - Flag indicating that the document is supported in the collaboration service
- **Data_Doc_IsOpeningOfflineCopy** - Flag indicating that the offline copy of a document was opened
- **Data_Doc_IsSyncBacked** - Flag indicating that an auto-synced copy of the document exists on the computer
- **Data_Doc_Location** - Indicates which service provided the document (OneDrive, File Server, SharePoint etc.)
- **Data_Doc_LocationDetails** - Indicates which Known Folder provided a locally stored document
- **Data_Doc_NumberCoAuthors** - Count of the number of fellow users in a collaborative editing session
- **Data_Doc_PasswordFlags** - Indicates read or read/write password flags set
- **Data_Doc_ReadOnlyReasons** - Reasons why the document was opened read only

- **Data_Doc_ResourceIdHash** - An anonymized document Identifier used to diagnose problems
- **Data_Doc_RtcType** - Indicates how the real-time channel (RTC) was set up for current file (Disabled, unsupported, on demand, always on, etc.).
- **Data_Doc_ServerDocId** - An immutable anonymized document Identifier used to diagnose problems
- **Data_Doc_ServerProtocol** - the protocol version used to communicate with the service
- **Data_Doc_ServerType** - the type of the server offering the service (SharePoint, OneDrive, WOPI, etc.)
- **Data_Doc_ServerVersion** - the server version offering the service
- **Data_Doc_SessionId** - Identifies a specific document edit session within the full session
- **Data_Doc_SharePointServiceContext** - Diagnostic information from SharePoint Online requests
- **Data_Doc_SizeInBytes** - Indicator of document size
- **Data_Doc_SpecialChars** - Indicator of special chars in the document's URL
- **Data_Doc_SyncBackedType** - Indicator as to the type of document (local or service based)
- **Data_Doc_UrlHash** - One-way hash to create a naïve document identifier
- **Data_Doc_WopiServiceId** - Contains unique identifier of WOPI service provider
- **Data_EditorDisablingRename** - Identifier of the first editor that caused for rename to be disabled
- **Data_EditorsCount** - Number of editors in the document
- **Data_ForceReadWriteReason** - Integer value representing the reason why the file was forced into read/write mode
- **Data_FSucceededAfterRecoverableFailure** - Indicates that open succeeded after repairing a failure while opening the document
- **Data_LastLoggedTag** - Unique tag for code call site used to identify when we try to fail the open twice (used for data quality diagnostics)
- **Data_LinkStyles** - Indicates whether we are linking to the template styles
- **Data_MainPdod** - The document identifier in Office Word process
- **Data_Measurements** - Encoded string containing the time breakdown of the different parts of open. Used to measure performance.
- **Data_MoveDisabledReason** - Error that is disabling move for the document
- **Data_MoveFlightEnabled** - Whether the flight for the move feature is enabled
- **Data_PartsUnknown** - the number of document parts that we couldn't get data for
- **Data_RecoverableFailureInitiationLocationTag** - Unique tag for code call site used to identify the place in code where we attempt to fix the file before opening it
- **Data_RenameDisabledReason** - Error that is causing for rename to be disabled for this document
- **Data_RenameFlightEnabled** - Whether the flight for the rename feature is enabled
- **Data_SecondaryTag** - Unique tag for code call site used to add additional failure data for open
- **Data_TemplateFormat** - File format of the template that the document is based on.
- **Data_UsesNormal** - Indicates whether the open document is based on the normal template

- **PathData_Doc_StreamAvailability** - Indicator if document stream is available/disabled

Office.Word.FileOpen.OpenFFileOpenXstzCore

This event indicates Office Word opens a document who is double-clicked by a user. It also contains critical file open performance data and is an app start event from user perspective. The event monitors whether file-openfrom-file-double-click is working as expected. It is also used to calculated monthly active users/devices, and cloud reliability metrics.

The following fields are collected:

- **Data_AddDocTelemRes** - Reports whether we were able to properly populate other document telemetry-related values in the event. Used for data quality diagnostics
- **Data_BytesAsynchronous** - Number of bytes (compressed) that we believe we can open the file without if we get them before the user wants to start editing or maybe saving
- **Data_BytesAsynchronousWithWork** - Number of bytes (compressed) that we might be able to open the file without but would require significant code investments to make it happen
- **Data_BytesSynchronous** - Number of bytes (compressed) that we must have before we can start opening the file
- **Data_BytesUnknown** - Number of bytes in document parts that we don't expect to find
- **Data_DetachedDuration** - How long was the activity detached from the thread
- **Data_Doc_AccessMode** - Document is read only/editable
- **Data_Doc_AssistedReadingReasons** - Predefined set of values of why document was opened in assisted reading mode
- **Data_Doc_AsyncOpenKind** – Indicates whether a cached version of the cloud document was opened and which asynchronous refresh logic was used.
- **Data_Doc_ChunkingType** - Units used for incremental document open
- **Data_Doc_EdpState** - Electronic Data Protection setting for the document
- **Data_Doc_Ext** - Document extension (docx/xlsb/pptx etc.)
- **Data_Doc_FileFormat** - File format protocol version
- **Data_Doc_Fqdn** - OneDrive or SharePoint Online Domain Name
- **Data_Doc_FqdnHash** - One-way hash of customer identifiable domain name
- **Data_Doc_IOFlags** - Reports on the cached flags used to set open request options
- **Data_Doc_IdentityTelemetryId** - A one-way hash of the user identity used to perform the open
- **Data_Doc_InitializationScenario** - Records how the document was opened
- **Data_Doc_IrmRights** - Actions permitted by the Electronic Data Protection policy that has been applied to the document/user
- **Data_Doc_IsIncrementalOpen** - Flag indicating that the document has been incrementally opened
- **Data_Doc_IsOcsSupported** - Flag indicating that the document is supported in the collaboration service
- **Data_Doc_IsOpeningOfflineCopy** - Flag indicating that the offline copy of a document was opened
- **Data_Doc_IsSyncBacked** - Flag indicating that an auto synced copy of the document exists on the

computer

- **Data_Doc_Location** - Indicates which service provided the document (OneDrive, File Server, SharePoint etc.)
- **Data_Doc_LocationDetails** - Indicates which Known Folder provided a locally stored document
- **Data_Doc_NumberCoAuthors** - Count of the number of fellow users in a collaborative editing session
- **Data_Doc_PasswordFlags** - Indicates read or read/write password flags set
- **Data_Doc_ReadOnlyReasons** - Reasons why the document was opened read only
- **Data_Doc_ResourceIdHash** - An anonymized document Identifier used to diagnose problems
- **Data_Doc_RtcType** - Indicates how the real-time channel (RTC) was set up for current file (Disabled, unsupported, on demand, always on, etc.).
- **Data_Doc_ServerDocId** - An immutable anonymized document Identifier used to diagnose problems
- **Data_Doc_ServerProtocol** - the protocol version used to communicate with the service
- **Data_Doc_ServerType** - the type of the server offering the service (SharePoint, OneDrive, WOPI etc.)
- **Data_Doc_ServerVersion** - the server version offering the service
- **Data_Doc_SessionId** - Identifies a specific document edit session within the full session
- **Data_Doc_SharePointServiceContext** - Diagnostic information from SharePoint Online requests
- **Data_Doc_SizeInBytes** - Indicator of document size
- **Data_Doc_SpecialChars** - Indicator of special chars in the document's URL or Path
- **Data_Doc_StreamAvailability** - Indicator if document stream is available/disabled
- **Data_Doc_SyncBackedType** - Indicator as to the type of document (local or service based)
- **Data_Doc_UrlHash** - One-way hash to create a naïve document identifier
- **Data_Doc_WopiServiceId** - Contains unique identifier of WOPI service provider
- **Data_EditorDisablingRename** - Identifier of the first editor that caused for rename to be disabled
- **Data_EditorsCount** - Number of editors in the document
- **Data_FSucceededAfterRecoverableFailure** - Indicates that open succeeded after repairing a failure while opening the document
- **Data_ForceReadWriteReason** - Integer value representing the reason why the file was forced into read/write mode
- **Data_LastLoggedTag** - Unique tag for code call site used to identify when we try to fail the open twice (used for data quality diagnostics)
- **Data_LinkStyles** - Indicates whether we are linking to the template styles
- **Data_MainPdod** - The document identifier in Office Word process
- **Data_Measurements** - Encoded string containing the time breakdown of the different parts of open. Used to measure performance.
- **Data_MoveDisabledReason** - Error that is disabling move for the document
- **Data_MoveFlightEnabled** - Whether the flight for the move feature is enabled

- **Data_PartsUnknown** - the number of document parts that we couldn't get data for
- **Data_RecoverableFailureInitiationLocationTag** - Unique tag for code call site used to identify the place in code where we attempt to fix the file before opening it.
- **Data_RenameDisabledReason** - Error that is causing for rename to be disabled for this document
- **Data_RenameFlightEnabled** - Whether the flight for the rename feature is enabled
- **Data_SecondaryTag** - Unique tag for code call site used to add additional failure data for open.
- **Data_TemplateFormat** - File format of the template that the document is based on.
- **Data_UsesNormal** - Indicates whether the open document is based on the normal template.

Office.Word.FileOpen.OpenIfrInitArgs

This event indicates Office Word opens a document via COM activation or command line. It also contains critical file open performance data, and is an app start event from user perspective. The event monitors whether file-open-from-command-line is working as expected. It is also used to calculated monthly active users/devices, and cloud reliability metrics.

The following fields are collected:

- **Data_AddDocTelemRes** - Reports whether we were able to properly populate other document telemetry-related values in the event. Used for data quality diagnostics.
- **Data_BytesAsynchronous** - Number of bytes (compressed) that we believe we can open the file without if we get them before the user wants to start editing or maybe saving.
- **Data_BytesAsynchronousWithWork** - Number of bytes (compressed) that we might be able to open the file without but would require significant code investments to make it happen
- **Data_BytesSynchronous** - Number of bytes (compressed) that we must have before we can start opening the file
- **Data_BytesUnknown** - Number of bytes in document parts that we don't expect to find.
- **Data_Doc_AccessMode** - Document is read only/editable
- **Data_Doc_AssistedReadingReasons** - Predefined set of values of why document was opened in assisted reading mode
- **Data_Doc_AsyncOpenKind** – Indicates whether a cached version of the cloud document was opened and which asynchronous refresh logic was used.
- **Data_Doc_ChunkingType** - Units used for incremental document open
- **Data_Doc_EdpState** - Electronic Data Protection setting for the document
- **Data_Doc_Ext** - Document extension (docx/xlsb/pptx, etc.)
- **Data_Doc_FileFormat** - File format protocol version
- **Data_Doc_Fqdn** - OneDrive or SharePoint Online Domain Name
- **Data_Doc_FqdnHash** - One-way hash of customer identifiable domain name
- **Data_Doc_IOFlags** - Reports on the cached flags used to set open request options
- **Data_Doc_IdentityTelemetryId** - A one-way hash of the user identity used to perform the open
- **Data_Doc_InitializationScenario** - Records how the document was opened

- **Data_Doc_IrmRights** - Actions permitted by the Electronic Data Protection policy that has been applied to the document/user
- **Data_Doc_IsIncrementalOpen** - Flag indicating that the document has been incrementally opened
- **Data_Doc_IsOcsSupported** - Flag indicating that the document is supported in the collaboration service
- **Data_Doc_IsOpeningOfflineCopy** - Flag indicating that the offline copy of a document was opened
- **Data_Doc_IsSyncBacked** - Flag indicating that an auto synced copy of the document exists on the computer
- **Data_Doc_Location** - Indicates which service provided the document (OneDrive, File Server, SharePoint)
- **Data_Doc_LocationDetails** - Indicates which Known Folder provided a locally stored document
- **Data_Doc_NumberCoAuthors** - Count of the number of fellow users in a collaborative editing session
- **Data_Doc_PasswordFlags** - Indicates read or read/write password flags set
- **Data_Doc_ReadOnlyReasons** - Reasons why the document was opened read only
- **Data_Doc_ResourceIdHash** - An anonymized document Identifier used to diagnose problems
- **Data_Doc_RtcType** - Indicates how the real-time channel (RTC) was set up for current file (Disabled, unsupported, on demand, always on, etc.).
- **Data_Doc_ServerDocId** - An immutable anonymized document Identifier used to diagnose problems
- **Data_Doc_ServerProtocol** - the protocol version used to communicate with the service
- **Data_Doc_ServerType** - the type of the server offering the service (SharePoint, OneDrive, WOPI etc.)
- **Data_Doc_ServerVersion** - the server version offering the service
- **Data_Doc_SessionId** - the server version offering the service
- **Data_Doc_SharePointServiceContext** - Diagnostic information from SharePoint Online requests
- **Data_Doc_SizeInBytes** - Indicator of document size
- **Data_Doc_SpecialChars** - Indicator of special chars in the document's URL or Path
- **Data_Doc_StreamAvailability** - Indicator if document stream is available/disabled
- **Data_Doc_SyncBackedType** - Indicator as to the type of document (local or service based)
- **Data_Doc_UrlHash** - One-way hash to create a naïve document identifier
- **Data_Doc_WopiServiceId** - Contains unique identifier of WOPI service provider
- **Data_EditorDisablingRename** - Identifier of the first editor that caused for rename to be disabled
- **Data_EditorsCount** - Number of editors in the document
- **Data_FSucceededAfterRecoverableFailure** - Indicates that open succeeded after repairing a failure while opening the document
- **Data_ForceReadWriteReason** - Integer value representing the reason why the file was forced into read/write mode
- **Data_LastLoggedTag** - Unique tag for code call site used to identify when we try to fail the open twice

(used for data quality diagnostics)

- **Data_LinkStyles** - Indicates whether we are linking to the template styles
- **Data_MainPdod** - The document identifier in Office Word process
- **Data_Measurements** - Encoded string containing the time breakdown of the different parts of open. Used to diagnose open performance.
- **Data_MoveDisabledReason** - Error that is disabling move for the document
- **Data_MoveFlightEnabled** - Whether the flight for the move feature is enabled
- **Data_PartsUnknown** - the number of document parts that we couldn't get data for
- **Data_RecoverableFailureInitiationLocationTag** - Unique tag for code call site used to identify the place in code where we attempt to fix the file before opening it
- **Data_RenameDisabledReason** - Error that is causing for rename to be disabled for this document
- **Data_RenameFlightEnabled** - Whether the flight for the rename feature is enabled
- **Data_SecondaryTag** - Unique tag for code call site used to add additional failure data for open.
- **Data_TemplateFormat** - File format of the template that the document is based on.
- **Data_UsesNormal** - Indicates whether the open document is based on the normal template.

Office.Word.FileOpen.OpenLoadFile

This event indicates Office Word opens a document via Open dialog. It also contains critical file open performance data and is an app start event from user perspective. The event monitors whether file-open-from-the-open-filedialog is working as expected. It is also used to calculated monthly active users/devices, and cloud reliability metrics.

The following fields are collected:

- **Data_AddDocTelemRes** - Reports whether we were able to properly populate other document telemetry-related values in the event. Used for data quality diagnostics.
- **Data_BytesAsynchronous** - Number of bytes (compressed) that we believe we can open the file without if we get them before the user wants to start editing or maybe saving
- **Data_BytesAsynchronousWithWork** - Number of bytes (compressed) that we might be able to open the file without but would require significant code investments to make it happen
- **Data_BytesSynchronous** - Number of bytes (compressed) that we must have before we can start opening the file
- **Data_BytesUnknown** - Number of bytes in document parts that we don't expect to find
- **Data_DetachedDuration** - How long was the activity detached from the thread
- **Data_Doc_AccessMode** - Document is read only/editable
- **Data_Doc_AssistedReadingReasons** - Predefined set of values of why document was opened in assisted reading mode
- **Data_Doc_AsyncOpenKind** – Indicates whether a cached version of the cloud document was opened and which asynchronous refresh logic was used.
- **Data_Doc_ChunkingType** - Units used for incremental document open
- **Data_Doc_EdpState** - Electronic Data Protection setting for the document

- **Data_Doc_Ext** - Document extension (docx/xlsb/pptx etc.)
- **Data_Doc_FileFormat** - File format protocol version
- **Data_Doc_Fqdn** - OneDrive or SharePoint Online Domain Name
- **Data_Doc_FqdnHash** - One-way hash of customer identifiable domain name
- **Data_Doc_IdentityTelemetryId** - A one-way hash of the user identity used to perform the open
- **Data_Doc_InitializationScenario** - Records how the document was opened
- **Data_Doc_IOFlags** - Reports on the cached flags used to set open request options
- **Data_Doc_IrmRights** - Actions permitted by the Electronic Data Protection policy that has been applied to the document/user
- **Data_Doc_IsIncrementalOpen** - Flag indicating that the document has been incrementally opened
- **Data_Doc_IsOcsSupported** - Flag indicating that the document is supported in the collaboration service
- **Data_Doc_IsOpeningOfflineCopy** - Flag indicating that the offline copy of a document was opened
- **Data_Doc_IsSyncBacked** - Flag indicating that an auto synced copy of the document exists on the computer
- **Data_Doc_Location** - Indicates which service provided the document (OneDrive, File Server, SharePoint etc.)
- **Data_Doc_LocationDetails** - Indicates which Known Folder provided a locally stored document
- **Data_Doc_NumberCoAuthors** - Count of the number of fellow users in a collaborative editing session
- **Data_Doc_PasswordFlags** - Indicates read or read/write password flags set
- **Data_Doc_ReadOnlyReasons** - Reasons why the document was opened read only
- **Data_Doc_ResourceIdHash** - An anonymized document Identifier used to diagnose problems
- **Data_Doc_RtcType** - Indicates how the real-time channel (RTC) was set up for current file (Disabled, unsupported, on demand, always on, etc.).
- **Data_Doc_ServerDocId** - An immutable anonymized document Identifier used to diagnose problems
- **Data_Doc_ServerProtocol** - the protocol version used to communicate with the service
- **Data_Doc_ServerType** - the type of the server offering the service (SharePoint, OneDrive, WOPI, etc.)
- **Data_Doc_ServerVersion** - the server version offering the service
- **Data_Doc_SessionId** - Identifies a specific document edit session within the full session
- **Data_Doc_SharePointServiceContext** - Diagnostic information from SharePoint Online requests
- **Data_Doc_SizeInBytes** - Indicator of document size
- **Data_Doc_SpecialChars** - Indicator of special chars in the document's URL or Path
- **Data_Doc_StreamAvailability** - Indicator if document stream is available/disabled
- **Data_Doc_SyncBackedType** - Indicator as to the type of document (local or service based)
- **Data_Doc_UrlHash** - One-way hash to create a naïve document identifier

- **Data_EditorDisablingRename** - Identifier of the first editor that caused for rename to be disabled
- **Data_EditorsCount** - Number of editors in the document
- **Data_ForceReadWriteReason** - Integer value representing the reason why the file was forced into read/write mode
- **Data_FSucceededAfterRecoverableFailure** - Indicates that open succeeded after repairing a failure while opening the document
- **Data_LastLoggedTag** - Unique tag for code call site used to identify when we fail to try the save twice (used for data quality diagnostics)
- **Data_LinkStyles** - Indicates whether we are linking to the template styles
- **Data_MainPdod** - The document identifier in Office Word process
- **Data_Measurements** - Encoded string containing the time breakdown of the different parts of open. Used to measure performance.
- **Data_MoveDisabledReason** - Error that is disabling move for the document
- **Data_MoveFlightEnabled** - Whether the flight for the move feature is enabled
- **Data_PartsUnknown** - the number of document parts that we couldn't get data for
- **Data_RecoverableFailureInitiationLocationTag** - Unique tag for code call site used to identify the place in code where we attempt to fix the file before opening it
- **Data_RenameDisabledReason** - Error that is causing for rename to be disabled for this document
- **Data_RenameFlightEnabled** - Whether the flight for the rename feature is enabled
- **Data_SecondaryTag** - Unique tag for code call site used to add additional failure data for open
- **Data_TemplateFormat** - File format of the template that the document is based on
- **Data_UsesNormal** - Indicates whether the open document is based on the normal template

RenewUserOp

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when renew user certificates operation is performed.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **iKey** - Logger server ID
- **RMS.ApplicationScenarioId** - Scenario ID provided by the application
- **RMS.Duration** - Total time for the operation to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the operation

- **RMS.HttpCall** - indicates if there is HTTP operation
- **RMS.LicenseFormat** - The license Format: Xrml or Json
- **RMS.Result** - Success or fail of the operation
- **RMS.ScenarioId** - Scenario ID defined by Rights Management Service Client
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.ServerType** - The type of Rights Management Service Server
- **RMS.StatusCode** - Status code of the operation result
- **RMS.Type** - The type of user info

ServiceDiscoveryOp

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when service discovery operation is performed.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.ApplicationScenarioId** - Scenario ID provided by the application
- **RMS.Duration** - Total time for the operation to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the operation
- **RMS.HttpCall** - Indicate if there is HTTP operation
- **RMS.LicenseFormat** - The license Format: Xrml or Json
- **RMS.OperationName** - Operation name
- **RMS.Result** - Success or fail of the operation
- **RMS.ScenarioId** - Scenario ID defined by Rights Management Service Client
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.ServerType** - The type of Rights Management Service Server
- **RMS.StatusCode** - Status code of the operation result

Office accessibility configuration subtype

Office accessibility features

Office.Accessibility.AccessibilityToolSessionPresenceWin32

Allows us to detect that the user has an Assistive technology tool and its name. This allows us to understand if an Office user is experiencing issues with a specific Assistive Technology tool.

The following fields are collected:

- **Data_Data_Jaws** - indicates if Jaws was running during the session**Data_Data_Magic** - indicates if Magic was running during the session
- **Data_Data_Magnify** - indicates if Magnify was running during the session
- **Data_Data_Narrator** - indicates if Narrator was running during the session
- **Data_Data_NVDA** - indicates if NVDA was running during the session
- **Data_Data_SA** - indicates if SA was running during the session
- **Data_Data_Supernova** - indicates if Supernova was running during the session
- **Data_Data_SuperNovaessSuite** - indicates if SuperNovaAccessSuite was running during the session
- **Data_Data_WinEyes** - indicates if WinEyes was running during the session
- **Data_Data_ZoomText** - indicates if ZoomText was running during the session

Office.Apple.DarkMode

This event is collected for Office applications running under Apple platforms. The event tells us if user is running a System on DarkMode and whether the user overwrote the DarkMode System setting in Office. We use this event to help ensure accessibility and prioritize user-experience optimization.

The following fields are collected:

- **Data_DarkModeIsEnabled** - Whether DarkMode is enabled in the system.
- **Data_RequiresAquaSystemAppearanceEnabled** - Whether DarkMode is overwritten in Office.

Office.Apple.HardwareKeyboardInUse.Apple

This event is collected for Office applications running under Apple platforms. The event tells us that a user is attaching a keyboard to their mobile device. The event helps us improve accessibility and optimize our user experience.

The following fields are collected:

- **Data_CollectionTime** - A timestamp denoting the event collection time.

Office.Apple.MbuInstrument.DeviceAccessibilitySettings

This event is collected for Office applications running under Apple platforms. The event collects the state of the different accessibility options available during a session. We use this event to help ensure accessibility and prioritize user-experience optimization.

The following fields are collected:

- **Data_AccessibilityContentSize** - A flag indicating whether this setting is enabled
- **Data_AssistiveTouchRunning** - A flag indicating whether this setting is enabled
- **Data_BoldTextEnabled** - A flag indicating whether this setting is enabled
- **Data_CollectionTime** - A flag indicating whether this setting is enabled
- **Data_DarkerSystemColorsEnabled** - A flag indicating whether this setting is enabled
- **Data_DifferentiateWithoutColor** - A flag indicating whether this setting is enabled
- **Data_GrayscaleEnabled** - A flag indicating whether this setting is enabled
- **Data_GuidedAccessEnabled** - A flag indicating whether this setting is enabled

- **Data_IncreaseContrast** - A flag indicating whether this setting is enabled
- **Data_InvertColorsEnabled** - A flag indicating whether this setting is enabled
- **Data_PreferredContentSizeCategory** - A flag indicating whether this setting is enabled
- **Data_ReduceMotionEnabled** - A flag indicating whether this setting is enabled
- **Data_ReduceTransparency** - A flag indicating whether this setting is enabled
- **Data_ReduceTransparencyEnabled** - A flag indicating whether this setting is enabled
- **Data_ShakeToUndeEnabled** - A flag indicating whether this setting is enabled. (Deprecated - used only on old builds.)
- **Data_ShakeToUndoEnabled** - A flag indicating whether this setting is enabled.
- **Data_SpeakScreenEnabled** - A flag indicating whether this setting is enabled
- **Data_SpeakSelectionEnabled** - A flag indicating whether this setting is enabled
- **Data_SwitchControlRunning** - A flag indicating whether this setting is enabled
- **Data_UAZoomEnabled** - A flag indicating whether this setting is enabled
- **Data_VoiceOverRunning** - A flag indicating whether this setting is enabled

Office.Word.Accessibility.LearningTools.ReadAloud.PlayReadAloud

This event indicates Office Word reads aloud the text in the document. The event is a heartbeat of the accessibility feature, which allows Microsoft to evaluate the feature health of read-aloud-text.

The following fields are collected:

- **Data_ParagraphCount** - paragraph count of the document
- **Data_Play** - Is this the first time for Word to read aloud
- **Data_ViewKind** - view type of the document

Office.Word.Accessibility.LearningTools.ReadAloud.StopReadAloud

This event indicates Office Word stops reading aloud the text in the document. The event allows Microsoft to evaluate the feature health of read-aloud-text by measuring the working duration.

The following fields are collected:

- None

Privacy subtype

Office privacy settings

Office.Android.DocsUI.Views.UpsellBlockedAADC

This event captures that subscription upsell was blocked for non-adult users when they encountered the upsell message in Word, Excel or PowerPoint mobile app. We will use the data to summarize how many upsell opportunities were lost due to age compliance.

The following fields are collected:

- **EntryPoint** – String specifying the upsell entry point which was blocked for age compliance.

Office.IntelligentService.PrivacyConsent.PrivacyEvent

This event represents a user or system initiated action that is part of the privacy User experience for Office. It is triggered on the privacy First Run dialogs, Account Privacy dialog, and privacy notifications. The event is used to understand the following: users consenting to Office privacy settings, users changing Office privacy settings, and

Office privacy settings getting updated in user sessions.

The following fields are collected:

- **Data_ActionId** - User action in a privacy dialog
- **Data_ControllerConnectedServicesState** - User policy setting for additional optional connected experiences
- **Data_DownloadedContentServiceGroupState** - User setting for downloaded content
- **Data_ForwardLinkId** - Link to privacy documentation for the user scenario
- **Data_HRESULT** - Record of errors during interaction with a privacy dialog
- **Data_IsEnterpriseUser** - User license category
- **Data_OfficeServiceConnectionState** - User setting for connected services
- **Data_RecordRegistry** - Record of showing the enterprise privacy dialog
- **Data_Scenario** - First run scenario based on the user license and category
- **Data_SeenInsidersDialog** - Record of showing the Insiders privacy dialog
- **Data_SendTelemetryOption** - User setting for telemetry
- **Data_SendTelemetryOptionPolicy** - User policy setting for telemetry
- **Data_UserCategory** - User account type
- **Data_UserCCSDisabled** - User override for additional optional connected experiences
- **Data_UserContentServiceGroupState** - User setting for analyzing content
- **Data_WillShowDialogs** - Record of user needing to see privacy First Run dialogs

Office.OfficeMobile.FRE.UpsellBlockedAADC

This event captures that subscription upsell was blocked for non-adult users when they encountered the upsell message in the Office Mobile app. We will use the data to summarize how many upsell opportunities were lost due to age compliance.

The following fields are collected:

- **EntryPoint** – String specifying the upsell entry point which was blocked for age compliance.

Product and service performance data events

The following are the data subtypes in this category:

- Unexpected application exit (crash)
- Application feature performance
- Application activity error

Unexpected application exit (crash) subtype

Unexpected application exits and the state of the application when that happens.

app.startup.reason

This event lets us detect and fix issues where Outlook crashed during app startup. This event includes information on why the crash happened so we can fix the issue quickly.

The following fields are collected:

- **app_background_time** - duration of how long app was in background last session
- **startup_reason_type** - indicates why the app is starting up, this will indicate if it was due to force quit, or other reason.
- **watch_status_info** - keeps track of the following information, if applicable.
  - **is_watch_app_installed** - determines if the user has the Watch app installed
  - **is_watch_paired** - determines if iOS device is paired to a watch
  - **is_watch_supported_and_active** - indicates whether a watch is supportive and active during the session

The following fields are collected for only Outlook Mobile for iOS:

- **clean_exit_reason** - A string of words indicating why if there was a reason for the app stop
- **is_agenda_user** - Indicates if the user has opened the agenda recently, which indicates if we are writing disk on the startup
- **is_watch_supported_and_active** - indicates whether a watch is supportive and active during the session

application.crash

Used for monitoring critical app crashes and helps us collect information on why the app has crashed and how to prevent it.

The following fields are collected:

- **android.hardware.** - (for example, android.hardware.bluetooth) Hardware configuration values provided by the Android platform
- **android.software.** - (for example, android.software.device_admin) Software configuration values provided by the Android platform
- **android_version** - Device android version name as indicated by android.os.Build.VERSION#RELEASE
- **application_package_name** - Application package name as indicated by android.content.Context#getPackageName()
- **application_stack_trace** - the stack trace of the crash
- **application_version_code** - Application version code defined by the Outlook app
- **application_version_name** - Application version name defined by the Outlook app
- **com.** (for example, com.google.android.feature.FASTPASS_BUILD, com.amazon.feature.PRELOAD, com.samsung.android.bio.face) Manufacturer-specific configuration values provided by the Android platform
- **crash_report_sdk** - SDK to send crash logs. Either Hockey or AppCenter
- **crash_type** - crash_type will have java, native, non-fatal as types.
  - java - if crash recorded on application Layer.
  - native - if crash recorded on native layer within the app.
  - non-fatal - crashes being recorded to debug any feature. Application won't crash but it will upload non-fatal crash logs to help in debugging a feature.
- **device_brand** - Device brand (manufacturer or carrier) as indicated by android.os.Build#BRAND

- **device_ID** - Device unique ID (IMEI) *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **device_manufacturer** - Device manufacturer as indicated by android.os.Build#MANUFACTURER
- **device_model** - Device model as indicated by android.os.Build#MODEL
- **device_name** - Device name as indicated by android.os.Build#DEVICE
- **device_total_memory** - Estimation of the total device memory size based on filesystem stats.
- **glEsVersion** - OpenGL Embedded Systems version key

crash.event

Allows us to detect and fix situations where critical app crashes have occurred and helps us collect information on why the app has crashed and how to prevent it.

The following fields are collected:

- **crashTime** - Date and time the crash occurred to help with investigation
- **crash_time_from_start** – The elapsed time from app start to the crash occurred, to help with investigation
- **exceptionName** - The name of the exception that triggered the crash to help with investigation
- **exception_reason** – The reason of the exception that triggered the crash to help with investigation
- **hasHx** - Tells us the account is using our new sync service to help us detect issues caused by our sync service
- **incidentIdentifier** - A unique ID for the crash report so we can find the corresponding issue
- **isAppKill** - Helps us understand if that app was killed or close on the device
- **is_crashloop** – Helps us understand if the crash could likely be a crash loop.
- **reportKey** - A unique ID for the application installation om the device for issue investigation
- **signal** - A signal that caused the crash to give us more details to investigate this crash

Error

Allows us to understand the issues that mobile apps are facing when attempting to fetch privacy settings from the server.

The following fields are collected:

- **correlationId** - a unique identifier for a service connection that resulted in an error, allowing us to diagnose what might have gone wrong
- **errorCode** - identifies the relevant error code received from the service that could be used to diagnose the problem
- **exceptionType** - type of error that the library encountered when fetching the setting
- **message** - identifies the error message received from the service
- **roamingSettingType** - identifies the location from which we attempt to read settings
- **settingId** - the setting that was attempted to be fetched

Office.AppDomain.UnhandledExceptionHandlerFailed

Collects information for any unhandled exceptions using the Data Streamer App. This data is used to monitor

the health of the application. This event is generated by Microsoft Data Streamer for Excel Add-in.

The following fields are collected:

- **Exception** - Call stack for the Exception
- **Event Name** - Event Name is the Event Category and Event Label.

Office.Apple.IdentityDomainName

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of our system as well as investigating causes of failures by certain domain users. We collect the domain used by our users when they authenticate. We use this data to help identify and fix those issues that might not seem too impactful at a broader level, but that turn out to be very impactful to a certain domain of users.

The following fields are collected:

- **Data_Domain** - the domain used for authentication
- **Data_IdentityProvider** - The authentication identity provider name. (that is, LiveId or ADAL)
- **Data_IdentityProviderEnum** - The authentication identity provider code. (A number)

Office.Apple.SystemHealthAppExitMacAndiOS

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of our Office applications and for investigating causes of failures. We collect data on each application exit to determine whether an application exited gracefully.

The following fields are collected:

- **Data_AffectedProcessSessionID** - The identifier for the session that experience the application exit.
- **Data_AffectedSessionBuildNumber** - The minor version of the application in which an application exit was observed.
- **Data_AffectedSessionDuration** - The length of the session from start to exit
- **Data_AffectedSessionIDSMatch** - An indicator of telemetry health.
- **Data_AffectedSessionMERPSessionID** - An indicator of telemetry health.
- **Data_AffectedSessionOSLocale** - The Locale of the OS under which the application exit was observed.
- **Data_AffectedSessionOSVersion** - The OS version under which the application exit was observed.
- **Data_AffectedSessionResidentMemoryOnCrash** - The amount of resident memory that was consumed when the application exit occurred
- **Data_AffectedSessionStackHash** - An identifier that will denote the specific error being hit.
- **Data_AffectedSessionStartTime** - The time at which the session started.
- **Data_AffectedSessionUAEType** - The type of application exit observed (if it was an ungraceful exit, this code will denote the type of error observed)
- **Data_AffectedSessionVersion** - The major version of the application in which an application exit was observed.
- **Data_AffectedSessionVirtualMemoryOnCrash** - The amount of virtual memory that was consumed when the application exit occurred.
- **Data_ExitWasGraceful** - Whether the application Exit was graceful or ungraceful.

Office.Extensibility.COMAddinUnhandledException

Event generated when COM Add-in crashes on a consumer version of Office applications.

This is used to compute a global, non-enterprise-specific Microsoft 365 Apps for enterprise "adoption" for an add-in, which is then used by tools like the Readiness Toolkit. This allows enterprise customers to validate if the add-ins they have deployed in their organizations are compatible with the latest versions of Microsoft 365 Apps for enterprise and plan their upgrades accordingly.

The following fields are collected:

- **ScopeId** – the current thread scope
- **Method** – Office method where exception occurred
- **Interface** – Office interface where exception occurred
- **AddinId** – the add-in Class ID
- **AddinProgId** – deprecated
- **AddinFriendlyName** – deprecated
- **AddinTimeDateStamp** – the add-in timestamp from the DLL metadata
- **AddinVersion** – deprecated
- **AddinFileName** – deprecated
- **VSTOAddIn** – whether add-in is VSTO
- **AddinConnectFlag** – current load behavior
- **LoadAttempts** – number of attempts to load add-in

Office.Extensibility.COMAddinUnhandledExceptionEnterprise

Event generated when COM Add-in crashes on an enterprise version of Office applications.

This is used to compute a global, non-enterprise-specific Microsoft 365 Apps for enterprise "adoption" for an add-in, which is then used by tools like the Readiness Toolkit. This allows enterprise customers to validate if the add-ins they have deployed in their organizations are compatible with the latest versions of Microsoft 365 Apps for enterprise and plan their upgrades accordingly.

- **ScopeId** – the current thread scope
- **Method** – Office method where exception occurred
- **Interface** – Office interface where exception occurred
- **AddinId** – the add-in Class ID
- **AddinProgId** – deprecated
- **AddinFriendlyName** – deprecated
- **AddinTimeDateStamp** – the add-in timestamp from the DLL metadata
- **AddinVersion** – deprecated
- **AddinFileName** – deprecated
- **VSTOAddIn** – whether add-in is VSTO
- **AddinConnectFlag** – current load behavior
- **LoadAttempts** – number of attempts to load add-in

Office.Extensibility.Sandbox.ODPActivationHeartbeat

Office Add-ins run in a sandbox. This event collects heartbeat information on activations. When an add-in crashes, this event collects why it crashed in the case it's related to our sandbox. Used to investigate when customers escalate issues.

The following fields are collected:

- **AppId** - ID of the App
- **AppInfo** - Data regarding the type of add-in (task pane or UILess or in content etc.) and the provider type (omen, SharePoint, filesystem etc.)
- **AppInstanceId** - ID of the app instance
- **AssetId** - Asset ID of the app
- **ErrorCode** - Total time spent
- **IsArm64** - indicates if the add-in activation is happening on an application emulated on an ARM64 device
- **IsAugmentationScenario** – indicates if the augmentation loop is responsible for initializing the Office Solutions Framework control
- **IsDebug** - indicates if session is a debug session
- **IsPreload** – indicates if the add-in is being preloaded in background for improving activation perf.
- **IsWdagContainer** – indicates if the add-in activation is being taken place in a Windows Defender Application Guard container.
- **NumberOfAddinsActivated** - Counter of add-ins activated
- **RemoterType** - Specifies the type of remoter (Trusted, untrusted, Win32webView, Trusted UDF, etc.) used to activate the add-in
- **StoreType** - Origin of the app
- **Tag**- Specifies where exactly the code has failed using the unique tag associated with it.
- **UsesSharedRuntime** - indicates the app uses sharedRuntime or not.

Office.Extensibility.VbaTelemetryBreak

Event generated when a macro-enabled file runs into a compile or runtime error

Desktop Analytics: This is used as numerator in the computation of enterprise-specific health status for macro types (for example, Word macros, Excel macros, etc.) which is used to infer during pilot if the add-in is "ready to upgrade" in the production ring.

The following fields are collected:

- **TagId** – the ID of the telemetry tag
- **BreakReason** – the reason for the break (runtime, compile, other error)
- **SolutionType** – type of solution (document, template, app add-in, COM add-in)
- **Data.ErrorCode** – error code reported by VBA engine

Office.FindTime.AppFailedToStart

Collected when app fails to start due to an unexpected error during startup. Used to track exceptions & crashes. Helps monitor and debug app health.

The following fields are collected:

- **DateTime** - Timestamp of when the event is logged.
- **EventName** - The name of the event being logged.

Office.Outlook.Desktop.HangBucketMetrics

Collects hang time for outlook hangs – a unique identifier per hang, elapsed time, and call stack information. The data is used to detect and identify app crashes in order to fix in future updates.

The following fields are collected:

- **BucketId** - hash of the call stack
- **ElapsedTotal** - total time spent in the call
- **ElapsedHanging** - hang time spent in the call

Office.Outlook.Desktop.HangReportingScopePerfMetrics

Collects time taken for outlook core scenarios – switchfolder, switchmodule, sendmailoutbox, openitemclassic, sendmailtransport. The data is actively monitored for anomalous performance issues. It is used to detect and diagnose performance issues and improve performance with each update.

The following fields are collected:

- **ElapsedTotal** - Total time spent in this call
- **ScopeId** - name of the function containing the declaration or a custom name given through scope definition

Office.Outlook.Desktop.WatsonBuckets

This rule collects the crash information from the event logs when Outlook crashed in the previous session.

The data is actively monitored for anomalous hangs. It is used to detect and identify hangs in order to fix in future updates.

The following fields are collected:

- **BucketId** – hash of the call stack
- **ElapsedTotal** - total time spent in the call
- **ElapsedHanging** - hang time spent in the call

Office.PowerPoint.Session

Collecting feature usages on each PowerPoint session. This data is used to calculate the ratio of PowerPoint ungraceful exit while using a feature. The ratio of PowerPoint ungraceful exit is a key signal to guarantee PowerPoint is running as expected.

The following fields are collected:

- **Flag** – session flags
- **Usage** – feature usages

Office.PowerPoint.UAE.Dialog

Collected every time when PowerPoint ungracefully exited while a dialog open on top of PowerPoint main window. This information is critical to catch PowerPoint ungracefully exits due to an active dialog blocking PowerPoint main window. Microsoft is using this data to diagnose the issue in order to guarantee PowerPoint running as expected.

The following fields are collected:

- **DlgCnt** – total number of open dialogs when session crashed
- **DlgId** – open dialog identifier
- **IdType** – open dialog identifier type
- **InitTime** – crashed session initialized time
- **SessionId** – crashed session identifier
- **TopId** – top dialog identifier
- **Version** – crashed session version

Office.PowerPoint.UAE.Document

Collecting what feature is being used on a document when PowerPoint ungracefully exited. These features include slide show, document open, save, edit, co-authoring, shutdown. This information is critical to catch PowerPoint ungracefully exits while using a feature. Microsoft is using this data to diagnose the issue in order to guarantee PowerPoint running as expected.

The following fields are collected:

- **CoauthFlag** – coauth usage flags
- **CommandFlag** – document modification flags
- **FileIOFlag** – file IO usage flags
- **InitTime** - crashed session initialized time
- **Location** – document location
- **ServerDocId** – server document identifier
- **SessionId** - crashed session identifier
- **UrlHash** – document URL hash
- **Usage** – feature usages
- **Version** - crashed session version

Office.PowerPoint.UAE.Open

Collecting every time when PowerPoint ungracefully exit while opening a document. This information is critical to catch PowerPoint ungracefully exits while opening a document. Microsoft is using this data to diagnose the issue in order to guarantee PowerPoint running as expected.

The following fields are collected:

- **FileUrlLocation** – document URL location
- **Flag** – open flags
- **InitTime** - crashed session initialized time
- **Location** - document location
- **OpenReason** – open reason
- **ServerDocId** – server document identifier
- **SessionId** - crashed session identifier
- **UrlHash** - document URL hash

- **Version** - crashed session version

Office.PowerPoint.UAE.Session

Collecting what feature has been used when PowerPoint session ungracefully exited. This information is critical to catch PowerPoint ungracefully exits. Microsoft is using this data to diagnose the issue in order to guarantee PowerPoint running as expected.

The following fields are collected:

- **Flag** – session flags
- **InitTime** - crashed session initialized time
- **PreviousSessionId** - crashed session identifier
- **Usage** – feature usages
- **Version** - crashed session version

Office.PowerPoint.UAE.Shutdown

Collecting PowerPoint ungracefully exit while shutting down. This information is critical to catch PowerPoint ungraceful exits while shutting down. Microsoft uses this data to diagnose the issue in order to guarantee PowerPoint runs as expected.

The following fields are collected:

- **InitTime** - crashed session initialized time
- **SessionId** - crashed session identifier
- **Stage** – shutdown stage
- **Version** - crashed session version

Office.PowerPoint.UAE.SlideShow

Collecting PowerPoint ungracefully exit while shutting down. This information is critical to catch PowerPoint ungraceful exits while shutting down. Microsoft uses this data to diagnose the issue in order to guarantee PowerPoint runs as expected.

The following fields are collected:

- **InitTime** - crashed session initialized time
- **Mode** – slide show mode
- **SessionId** - crashed session identifier
- **State** – slide show state
- **Version** - crashed session version

Office.Programmability.Addins.COMAddInCrash

Event generated when a COM Add-in crashes. Used to determine adoption and reliability issues with Office add-ins.

The following fields are collected:

- **AddinConnectFlag** - Represents load behavior
- **AddinDescriptionV2** - Description of the add-in
- **AddinFileNameV2** - Name of the actual add-in DLL. Does not include file location.
- **AddinFriendlyNameV2** - Add-in friendly name

- **AddinIdV2** - Add-in class ID (CLSID)
- **AddinProgIdV2** - Add-in prog ID
- **AddinProviderV2** - Provider of the add-in
- **AddinTimeDateStampV2** - Compiler timestamp
- **AddinVersionV2** - Add-in version
- **Interface** - COM interface of add-in that led to crash
- **LoadAttempts** - how many load attempts were made prior to crash
- **Method** - COM method of add-in, which led to crash
- **OfficeArchitecture** - Architecture of the Office client

Office.Programmability.Telemetry.AddInCrash

Event generated when a COM Add-in is loaded. This information is critical to determine whether an add-in caused an Office application crash. It is used to assess global add-in compatibility with Office applications.

The following fields are collected:

- **CLSID** – the add-in Class identifier
- **ConnectFlag** – the current add-in load behavior
- **FileName** – the add-in file name, excluding the file path
- **FriendlyName** – the add-in friendly name
- **Interface** – the Office interface where the exception occurred
- **LoadAttempts** – the number of attempts to load the add-in
- **Method** – the Office method where the exception occurred
- **OfficeApplication** – the Office application where the exception occurred
- **OfficeVersion** – the versions of Office
- **ProgID** – the add-in Prog identifier

Office.Programmability.Telemetry.MacroFileOpened

Triggered upon opening a macro (VBA)-containing file on a device that has been onboarded to Office Apps as a Service (OAAS) by the IT admin and where Microsoft 365 Apps for enterprise has been activated with an enterprise license. The event is used to understand the health of macro(VBA)-containing files in a tenant and is compared to Office.Programmability.Telemetry.VbaTelemetryBreak which tracks errors on VBA-containing files.

No fields are collected.

Office.System.SystemHealthUngracefulAppExitMacAndiOS

On boot event that captures ungraceful app exits for further investigation.

The following fields are collected:

- **AffectedProcessAppBuild** – Build number
- **AffectedProcessAppBuildRevision** – Build Revision Number
- **AffectedProcessAppMajorVer** – Major version number of App
- **AffectedProcessAppMinorVer** – Minor version number of App

- **AffectedProcessAppName** – Application name
- **AffectedProcessResidentMemoryOnCrash** – Resident memory of crashed app
- **AffectedProcessUnsymbolicatedChecksum** – Goes with Stack hash for symbolization
- **AffectedProcessVirtualMemoryOnCrash** – Virtual memory of crashed app
- **AffectedSessionDuration** – Duration of session in seconds before crash
- **AffectedSessionLongBuildNumber** – Long build number
- **CrashedProcessSessionID** – SessionID of the process in app crash
- **DetectionTime** – DateTime of app crash
- **DeviceModel** – Hardware model
- **MERPSessionID** – Session ID of MERP
- **ReportingOsLocaleTag** – OS locale
- **ReportingOSVerStr** – OS version
- **SessionBuildNumber** – Crashed App's version
- **SessionIDSMatch** – Boolean to verify whether reporting session ID is the same as picked up by Merp
- **SessionVersion** – Crashed App's version– **StackHash** – Hash of the crashed app's stack trace
- **UAEType** – Enum giving us information on what type of crash it was

Office.ThisAddIn.StartupFailed

Collects information for exception that occur during startup of the Data Streamer App. This data is used to monitor the health of the application. This event is generated by Microsoft Data Streamer for Excel Add-in.

The following fields are collected:

- **Exception** - Call stack for the Exception
- **Event Name** - Event Name is the Event Category and Event Label.

OneNote.SafeBootAction

This is triggered during application start if the app crashed in the previous session. This data is used to track the new crashes and will help us identify if the crash detection logic is working properly and to keep track of number of boot crashes and early crashes.

The following fields are collected:

- **ActionType** - Possible values - IncrementCount, ResetBootCounter, ResetEarlyCounter
- **IsLoopCrash** - Possible values – Yes/No
- **IsNativeCrash** - Possible values - Yes/No

OneNote.SafeBootResetCrashCounterOnAppSuspend, Office.OneNote.Android.SafeBootResetCrashCounterOnAppSuspend, Office.Android.EarlyTelemetry.SafeBootResetCrashCounterOnAppSuspend

The critical signal is sent when we are resetting the crash counter on app suspend before safe boot dialog is shown. This marker is required to track and diagnose the health of the app. A safe boot dialog is shown when the app crashes multiple times continuously. It gives the user an option to reset the app. This marker will help figure out if Safe boot dialog was not shown to a user despite hitting trigger criteria.

The following fields are collected:

- None

telemetry.error

This event lets us diagnose and fix issues that are preventing necessary diagnostic data from being generated or sent. These events let us understand if we are missing critical data needed to identify security issues or major issues with how your app is working.

The following fields are collected:

- **timer_name** - Tells where the telemetry issue is happening, for example, in the mailbox component or the calendar. This helps us detect and resolve telemetry issues happening from a specific part of the app
- **type** - tells us the type of timer error to help us detect when our app is having any issues with sending diagnostic telemetry data

watchdog.anr

Needed for monitoring app performance errors to prevent cases where the app stops responding, and your screen becomes frozen in the app (referred to as ANR - application not responding).

The following fields are collected:

- **callstack** - the code callstack where the ANR occurred
- **caused_restart** - whether the app was forced to restart because of the ANR
- **duration** - the amount of time the device was frozen
- **id** - a unique identifier for the ANR
- **interval** - the configured threshold for triggering an ANR
- **is_application_object_initialized** - whether the ANR happened after the app was fully initialized or before
- **last_known_is_in_foreground** - whether the app was most recently in the foreground or background

Application feature performance subtype

Poor response time or performance for scenarios such as application startup or opening a file.

android.frame.metrics

Allows us to detect and fix situations where our Android app components are causing performance issues, for example, if your inbox is not scrolling smoothly.

The following fields are collected:

- **animation_duration** - duration of animation rendering in milliseconds
- **command_issue_duration** - duration to issue commands to the platform in milliseconds
- **draw_duration** - duration of drawing the UI in milliseconds
- **input_handling_duration** - duration of input handling in milliseconds
- **layout_measure_duration** - duration of measuring the layout in milliseconds
- **origin** - the app component that is being measured, for example calendar or mail
- **sync_duration** - duration to sync the frame in milliseconds
- **swap_buffers_duration** - duration to swap buffers in milliseconds
- **total_duration** - total duration of the frame rendering in milliseconds

- **unknown_delay** - delay caused by unknown sources other than the explicitly tracked durations

cal.component

This event lets us detect and fix issues where there is perceivable performance impact on our calendar UI components that would cause your calendar to have scrolling issues.

The following fields are collected:

- **above_40fps** - count of frames rendered above 40 fps
- **above_40rate** - radio of frames rendered above 40 fps
- **above_50fps** - count of frames rendered above 50 fps
- **above_50rate** - radio of frames rendered above 50 fps
- **above_55fps** - count of frames rendered above 55 fps
- **above_55rate** - radio of frames rendered above 55 fps
- **account_counter** - tracks the number of accounts associated for each type of calendar, for example, 2 for Gmail calendar and whether that account is using our new sync service
- **app_instance** – Outlook has two entry points for Duo, one is for Calendar and one is for Mail and both can be launched side by side in multi instance environment. This will let us know which instance is making this reporting call, either Mail or Calendar
- **component_name** - Tells us the name of the calendar component such as Agenda view or Day view to help us detect performance issues impacting a specific component in the calendar
- **display_frame_data** - Tracks the time spent on displaying every 60 frames to determine if there are performance issues.
- **orientation** - Tells us whether the device was in portrait or landscape mode to help us detect performance issues impacting a specific device orientation
- **taskId** – TaskId will give us the current instance's taskId. This will be required in multi instance environment if user wants to launch same instances (Calendar, Calendar or Mail, Mail) side by side
- **view_duration** - Tells us how long it took to render the various UI calendar components to help us detect performance issues impacting your calendar experience

contact.action

This event is triggered on different actions on contacts - viewing, updating, and deleting contacts as well as viewing the contacts list. It is used to determine if there are any performance regressions that have to do with contacts.

The following fields are collected:

- **accounts_with_filters** - the number of accounts with filters applied to the contact list
- **action** - the action that was performed, for example, viewing a contact
- **duration_initial_view_load** - duration from opening the view to initially loading the contact list
- **duration_show_contacts** - duration from opening the view to showing contacts in the contact list
- **total_contacts** - number of contacts with no filters applied
- **total_filtered_contacts** - number of contacts with filters applied

conversation.load.time

This event lets us detect and fix issues where there is perceivable performance impact on loading your email

conversations to ensure your emails are loading as expected.

The following fields are collected:

- **time** - Tells us the amount of time that it has taken for the email conversation to complete loading.

conversation.reloaded

This event lets us detect how often we reload the conversation based on service notifications. We need to track if the update notifications are being too loud and need to be trimmed because they are degrading usability.

The following fields are collected:

- **average** - the amount of reloads divided by the size
- **client-request-ID** - the client request identifier for the request that caused the error
- **date** - the date stamp of the request that caused the error
- **duration** - the time the conversation was open

core.data.migration

Allows us to detect and fix situations where there was an error in updating email data on your device to a newer version.

The following fields are collected:

- **db_size_megabytes** - tracks the size of the core data database rounded to the nearest 25 megabytes and with a maximum megabyte of 500
- **db_wal_size_megabytes** - tracks the size of the core data database when the main store file is untouched rounded to the nearest 1 megabyte and with a maximum megabyte of 10
- **free_space_megabytes** - tracks the free space available in buckets 10, 100, 1000, 10,000, and then 100,000.
- **migration_duration_seconds** - tracks the migration duration rounded to one of these time slots - 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180 (180 and beyond should just be 180)

core.data.performance

Allows us to detect and fix situations where the email data we're storing on your device is causing performance issues.

The following fields are collected:

- **Caller** - tracks the entity name that calls the save operation
- **db_size_megabytes** - tracks the size of the core data database rounded to the nearest 25 megabytes and with a maximum megabyte of 500
- **duration** - tracks the amount of time it takes to complete the operation
- **entity** - tracks the entity name that called the fetch operation
- **operation** - raw value of operation either save, fetch, or "read write queue blocked"

inbox.component

This event collects two types of user data: Microsoft 365 subscription status, and whether the user sees ads. This helps us detect and fix issues where there is perceivable performance impact on the user's inbox UI components, which would cause email messages, avatar, read/unread state to not load or display properly.

The following fields are collected:

- **above_40fps** - count of frames rendered above 40 fps
- **above_40rate** - radio of frames rendered above 40 fps
- **above_50fps** - count of frames rendered above 50 fps
- **above_50rate** - radio of frames rendered above 50 fps
- **above_55fps** - count of frames rendered above 55 fps
- **above_55rate** - radio of frames rendered above 55 fps
- **account_counter** - count of each account type present on the device, for example, an Office 365 account = 1 account, Outlook.com account = 1 account.
- **ad_not_shown_reason** - reason why ads are not being shown
- **ad_shown** - whether an ad was shown (if ads are enabled)
- **ad_shown_for_premium** - unexpectedly showing ad to premium users
- **age** - age of the person (used to confirm compliance with age limitations on ads) *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **app_instance** – Outlook has two entry points for Duo, one is for Calendar and one is for Mail and both can be launched side by side in multi instance environment. This will let us know which instance is making this reporting call, either Mail or Calendar
- **component_name** - the name of the component/view that is active during the filtering
- **has_hx** - whether the device has at least one Hx (our new email syncing service) account
- **has_subscription** - whether the device has an ads subscription
- **is_all_accounts_inbox** - whether the current inbox is the "all accounts" folder
- **is_current_account** - whether the current active account is the ads account
- **load_error_code** - error code when loading ads
- **network_error_code** - network error code when requesting ads
- **orientation** - the screen orientation at the time of the even (portrait or landscape)
- **provider** – the provider (Xandr or Facebook) of the current showing ad
- **sub_error_type** - detailed error type
- **taskId** – TaskId will give us the current instance's taskId. This will be required in multi instance environment if user wants to launch same instances (Calendar, Calendar or Mail, Mail) side by side
- **total_count** - total frames displayed by the component
- **view_duration** - how long the component was viewed by the user

Initial.page.landing

This event helps track the type of experience that users see when they land in our application page. This data is used to determine the traffic of users piped into each experience in our application and also helps us to easily consolidate experimentation results.

The following fields are collected:

- **Page** - This is used to track the type of experience that user first sees when they land on our page. Possible values are "Trial", "Skip", "Prebundled", "Subscription" etc.

- **storeExperience** - This is used to determine if user was eligible to see the Store SDK Experience.
- **stringVariant** - This is used to determine the type of strings that user sees when they land on our page. Note that for any page such as "Trial", user can be eligible to see different strings based on whether they had Legacy Office installed, or if they had previously activated Office. Possible enumerations of this property are "LegacyUpsell", "OfficeOpened", "Default", "YesIntent", "NoIntent" etc.
- **windowsBuildType** - This is used to track the type of WindowsBuildType that user is on. that is "RS4", "RS5", "RS19H1", "Vibranium etc. As our experiences are usually targeted to different WindowsBuildTypes, this property is vital in differentiating between rollouts.

IpcpBootstrapUser

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when the IpcpBootstrapUser API call is made.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.ApplicationScenarioId** - Scenario ID provided by the application
- **RMS.AuthCallbackProvided** - Indicate if provides the authentication callback as input of the API call or not
- **RMS.ConnectionInfo.ExtranetUrl** - the extranet URL in connection info
- **RMS.ConnectionInfo.IntranetUrl** - the intranet URL in connection info
- **RMS.ConnectionMode** - The connection mode between Rights Management Service client and server: online or offline
- **RMS.Duration** - Total time for API call to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the API call
- **RMS.GuestTenant** - Guest tenant ID for the user
- **RMS.HomeTenant** - Home tenant ID for the user
- **RMS.HttpCall** - indicate if there is HTTP operation
- **RMS.Identity.ExtranetUrl** - The extranet URL of Rights Management service server for the user, collected while getting a new Rights Account Certificate from the server
- **RMS.Identity.IntranetUrl** - The intranet URL of Rights Management service server for the user, collected while getting a new Rights Account Certificate from the server
- **RMS.Identity.Status** - The first time to get Rights Account Certificate from the server or renew the Rights Account Certificate
- **RMS.Identity.Type** - The type of the user account such as windows account or live account

- **RMS.Identity.UserProvided** - Indicate if the user email address provided or not while getting new Rights Account Certificate from the server
- **RMS.IssuerId** - The ID of the Rights Management Service server that issues Rights Account Certificate
- **RMS.LicenseFormat** - The license Format: Xrml or Json
- **RMS.RACType** - The type of Rights Accounts Certificate
- **RMS.Result** - Success or fail of the API call
- **RMS.ScenarioId** - Scenario ID defined by the API
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.ServerType** - The type of Rights Management Service Server
- **RMS.StatusCode** - Status code of the returned result
- **RMS.TemplatesCount** - The number of the templates
- **RMS.TokenProvided** - Indicate if provides the token as input of the API call or not
- **RMS.UserProvided** - Indicate if provides the consumer as input of the API call or not
- **UserInfo.UserObjectId** - The user object ID

IpcpGetKey

Collected when a user attempts to open an Information Rights Managed (IRM) protected document or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when the IpcpGetKey API call is made.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **iKey** - Logging service server ID
- **RMS.ApplicationScenarioId** - Scenario ID provided by the application
- **RMS.AuthCallbackProvided** - Indicate if provides the authentication callback as input of the API call or not
- **RMS.ConnectionMode** - The connection mode between Rights Management Service client and server: online or offline
- **RMS.ContentId** - Content ID of the document
- **RMS.Duration** - Total time for API call to complete
- **RMS.DurationWithoutExternalOps** - Total time minus external operations consumed, such as network latency.
- **RMS.ErrorCode** - The error code returned if any from the API call
- **RMS.EulId** - The ID of End User License
- **RMS.EulProvided** - Indicate if provides the End User License as input of the API call or not

- **RMS.GuestTenant** - Guest tenant ID for the user
- **RMS.HomeTenant** - Home tenant ID for the user
- **RMS.HttpCall** - indicate if there is http operation
- **RMS.Identity.ExtranetUrl** - The extranet URL of Rights Management service server for the user, collected while getting a new Rights Account Certificate from the server
- **RMS.Identity.IntranetUrl** - The intranet URL of Rights Management service server for the user, collected while getting a new Rights Account Certificate from the server
- **RMS.Identity.Status** - The first time to get Rights Account Certificate from the server or renew the Rights Account Certificate
- **RMS.Identity.Type** - The type of the user account such as windows account or live account
- **RMS.Identity.UserProvided** - Indicate if the user email address provided or not while getting new Rights Account Certificate from the server
- **RMS.IssuerId** - The ID of the Rights Management Service server that issues Rights Account Certificate
- **RMS.KeyHandle** - The memory address of key handle
- **RMS.LicenseFormat** - The license Format: Xrml or Json
- **RMS.PL.ExtranetUrl** - The extranet URL in Publishing License
- **RMS.PL.IntranetUrl** - The intranet URL in Publishing License
- **RMS.PL.KeyType** - Values of "Single" or "Double" Indicates whether the PL was protected with Single Key Protection or Double Key Protection
- **RMS.RACType** - The type of Rights Accounts Certificate
- **RMS.Result** - Success or fail of the API call
- **RMS.ScenarioId** - Scenario ID defined by the API
- **RMS.SDKVersion** - The version of Rights Management Service Client
- **RMS.ServerType** - The type of Rights Management Service Server
- **RMS.StatusCode** - Status code of the returned result
- **RMS.TemplatesCount** - The number of the templates
- **RMS.TokenProvided** - Indicate if provides the token as input of the API call or not
- **RMS.UserProvided** - Indicate if provides the consumer as input of the API call or not
- **UserInfo.UserObjectId** - The user object ID

json.parse.error

This event denotes that an error is thrown by the json parser. We will be able to debug the read registry string that was sent to the json parser, to allow a smooth experience for our users.

The following fields are collected:

- **Error** - This consists of the error message that the error object returns.

mail.filter.component

This event lets us detect and fix issues where there is perceivable performance impact on your mail filtering experience, which would cause your filters to not load or display properly.

The following fields are collected:

- **above_40fps** - count of frames rendered above 40 fps
- **above_40rate** - radio of frames rendered above 40 fps
- **above_50fps** - count of frames rendered above 50 fps
- **above_50rate** - radio of frames rendered above 50 fps
- **above_55fps** - count of frames rendered above 55 fps
- **above_55rate** - radio of frames rendered above 55 fps
- **account_counter** - count of each account type present on the device, for example, an Office 365 account = 1 account, Outlook.com account = 1 account.
- **ad_not_shown_reason** - reason why ads are not being shown
- **ad_shown** - whether an add was shown (if ads enabled)
- **age** - age of the person (used to confirm compliance with age limitations on ads)
- **app_instance** – Outlook has two entry points for Duo, one is for Calendar and one is for Mail and both can be launched side by side in multi instance environment. This will let us know which instance is making this reporting call, either Mail or Calendar
- **component_name** - the name of the component/view that is active during the filtering
- **folder_type** - the type of folder that is being filtered (for example, Inbox, Trash, NonSystem)
- **has_hx** - whether the device has at least one Hx (the new email syncing service) account
- **has_subscription** - whether the device has an ads subscription
- **is_all_accounts_inbox** - whether the current inbox is the "all accounts" folder
- **is_current_account** - whether the current active account is the ads account
- **load_error_code** - error code when loading ads
- **network_error_code** - network error code when requesting ads
- **orientation** - the screen orientation at the time of the even (portrait or landscape)
- **sub_error_type** - detailed error type
- **taskId** – TaskId will give us the current instance's taskId. This will be required in multi instance environment if user wants to launch same instances (Calendar, Calendar or Mail, Mail) side by side
- **total_count** - total frames displayed by the component
- **view_duration** - how long the component was viewed by the user

message.rendering.intercepted

This event enables us to track how often the users intercept the rendering process before it is completed. We use this data to detect performance issues.

The following fields are collected:

- **is_cache** - whether the message body is loaded from cache
- **is_on_screen** - whether the rendering process is visible to users (normal rendering)
- **is_rendering_complete** - whether the rendering process is completed

- **is_trimmed_body** - whether the message body is trimmed body
- **rendering_method** - the method of rendering message
- **rendering_time** - the duration of rendering the message until user leaves the page

message.rendering.performance

This event lets us monitor the performance of the message rendering process, so that we can analyze the performance of different rendering processes and detect performance issues.

The following fields are collected:

- **bundle_prepare_time** - the time to prepare the bundle for rendering
- **full_rendering_time** - the time of full rendering process
- **is_cache** - whether the message body is loaded from cache
- **is_on_screen** - whether the rendering process is visible to users (normal rendering)
- **is_trimmed_body** - whether the message body is trimmed body
- **load_message_time** - the time to load message from backend (can be 0 if the message has been cached)
- **native_preprocess_time** - the time to preprocess message body in native side
- **prepare_body_time** - the time to prepare message body (including load and preprocess message)
- **rendering_method** - the method of rendering message
- **rendering_time** - the time to render message by the bundle
- **wait_time** - the time to build message URL

Office.Android.AndroidOfficeLaunchToLandingPageLatency

Critical to capture for app performance metric with respect to the response time of the app from the boot. Microsoft uses this to collect the time taken for the app to be responsive and also detect scenarios that may impact boot time in Word, Excel, or PowerPoint.

The following fields are collected:

- **AnyCrashInteractionDuringBoot** - Boolean for any crash encountered during boot
- **AppActivationTimeInMs** - App phase time
- **AppSuspendedDuringBoot** - Boolean for app suspension during boot
- **AvailableMemoryInMB** - available memory
- **CollectionTime** - time of event
- **DalvikHeapLimitInMB** - Heap info
- **DocumentRecoveryInvoked** - Boolean to indicate if any document was recovered
- **ExtractionDone** - Native library extraction time
- **FastBootGainTimeInMs** - Time for fast boot completion
- **FileActivationAttempted** - Boolean to indicate if app was launched due to File activation
- **HasLogcatLoggingImpactOnBoot** - Boolean to indicate if logcat impacted boot time
- **IsThisFirstLaunch** - Boolean to indicate if this is first app launch

- **LatencyTimeInMilliSec** - latency in millisecond
- **LibrarySharingTimeInMs** - time for sharing libraries
- **LoadMinLibsTimeInMs** - Loading time for the minimum set of libraries
- **MruListingTimeInMs** - time to load MRU
- **NativeLibrariesLoadTime** - CPP library load time
- **NumberOfRunningProcesses** - number of running processes
- **NumberOfRunningProcesses** - Number of running processes
- **NumberOfRunningServices** - Number of running services
- **OfficeActivityTimeInMs** - Time for initing OfficeActivity
- **PostAppInitTimeInMs** - app phase time
- **PreAppInitializationTime** - App phase init time
- **PreAppInitTimeInMs** - app phase time
- **TotalMemoryInMB** - total memory
- **UIRaaSDownloadLanguagePackageBoot** - information related to language pack download
- **UserDialogInterruptionDuringBoot** - Boolean for any blocking dialog shown during boot

Office.Android.DocsUI.Views.DimePurchaseFlowState

This health event attempts to capture each state that a user goes through when the user is attempting to make a purchase through the in-app purchase flow hosted by Dime. The data is used to monitor and alert on the health of the purchase flow triggered from Office Mobile app when user opts to buy an Microsoft 365 subscription.

The following fields are collected:

- **EntryPoint** - Entry point of the purchase attempted by the user
- **OEMPreInstalled** - Identifies if the app is pre-installed or organically installed by the user
- **PurchaseState** - State of the user when attempting a purchase
  - 0 - Unknown error
  - 1 - Dime is attempted by the user for open
  - 2 - Network error
  - 3 - Dime is shown to the user
  - 4 - Dime is cancelled by the user
  - 5 - Refresh needed as purchase is successful
  - 6 - Purchase is successful
  - 7 - Generic dime error
  - 8 - Dime telemetry cannot be uploaded due to communication failure
  - 9 - Two instances of Dime running causing interruption error
  - 10 - Base WebURL loaded on officemobile app is invalid
  - 11 - Communication of officemobile app with Dime failed
  - 12 - No communication channel could be established
  - 13 - Communication ID could not be sent to Dime
  - 14 - The officemobile app is communicating to the wrong endpoint
  - 15 - AuthToken is not obtained for this MSA account

    - 16 - AuthToken is not sent to Dime
- **WebViewShownDuration** - Duration for which the dime purchase page is shown to the user

Office.Apple.Apple.AppBoot.Mac

This event is collected for Office applications running under Apple platforms. The event is used to collect the time taken to boot the app, as well as some details on the type of boot done. This event helps us monitor our performance and provide performance improvements.

The following fields are collected:

- **Data_ Data_EvtBootTimerDocStageReady** - Time elapsed until reaching certain point in code.
- **Data_DocumentRecoveryInvoked** - Whether document recovery was invoked during boot.
- **Data_EvtBootTimerBootIdle** - Time elapsed until reaching certain point in code.
- **Data_EvtBootTimerFinishLaunchEnd** - Time elapsed until reaching certain point in code.
- **Data_EvtBootTimerLaunchDidFinish** - Time elapsed until reaching certain point in code.
- **Data_EvtBootTimerLaunchStart** - Time elapsed until reaching certain point in code.
- **Data_EvtBootTimerMainStart** - Time elapsed until reaching certain point in code.
- **Data_EvtBootTimerStaticInit** - Time elapsed until reaching certain point in code.
- **Data_EvtDockStageReady** - Time elapsed until reaching certain point in code.
- **Data_IsFileOpenAttempted** - Whether a file open was attempted during boot.
- **Data_IsFirstRunAttempted** - Whether the app boot went through first run experience.
- **Data_SentToBackground** - Whether the app was sent to background during boot.

Office.Apple.DiskRuleResultSerializerErrorOnStreamOp

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of our telemetry infrastructure. This event denotes an error has occurred.

The following fields are collected:

- **Data_ActualBytesModified** - Number of bytes modified.
- **Data_BytesRequested** - Number of bytes to process.
- **Data_IsWriteOp** - Whether we are about to execute a write operation

Office.Apple.MacBootResourceUsage

This event is collected for Office applications running under Apple platforms. This event is collected for Office applications running under Apple platforms. The event is used to collect multiple indicators around the resources being consumed during boot by Office apps. This event helps us monitor our performance and provide performance improvements.

The following fields are collected:

- **Data_BlockInputOperations** - A count of block input operations
- **Data_BlockOutputOperations** - A count of block output operations
- **Data_InvoluntaryContextSwitches** - The number of involuntary context switches
- **Data_MainThreadCPUTime** - An elapsed time measurement
- **Data_MaxResidentSize** - A memory size measurement

- **Data_MessagesReceived** - The number of messages received
- **Data_MessagesSent** - The number of messages sent
- **Data_PageFaults** - A count of page reclaims
- **Data_PageReclaims** - A count of page reclaims
- **Data_ProcessCPUTime** - An elapsed time measurement
- **Data_SharedTextMemorySize** - A memory size measurement
- **Data_SignalsReceived** - The number of signals received
- **Data_Swaps** - A count of data swaps
- **Data_SystemCPUTime** - An elapsed time measurement
- **Data_SystemUpTime** - An elapsed time measurement
- **Data_UnsharedDataSize** - A data size measurement
- **Data_UnsharedStackSize** - A stack size measurement
- **Data_UserCPUTime** - An elapsed time measurement
- **Data_VoluntaryContextSwitchesNvcsw** - The number of voluntary context switches

Office.Apple.MAU.Validation

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of the Microsoft Autoupdate component, which is used to distribute and install application updates. The data collected is used for detecting errors and investigating causes of failures.

The following fields are collected:

- **Data_EventID** - We collect a string representing an error code
- **Data_Message** - We collect a string containing a description of the error

Office.Apple.MbuInstrument.Hang.Detection.Spin.Control

This event is collected for Office applications running under Apple platforms. The event is logged whenever an application appears to become non-responsive. This event helps us monitor our performance and provide performance improvements.

The following fields are collected:

- **Data_CountSpinControlStart** - A marker that indicates that the application appears to have become unresponsive (or slow to respond)

Office.Apple.MbuInstrument.VMOnDocumentClose

This event is collected for Office applications running under Apple platforms. The event is used to collect a snapshot of the state of memory during document close. This event helps us monitor our performance and provide performance improvements.

The following fields are collected:

- **Data_CollectionTime** - A timestamp from the moment in which the data was collected
- **Data_ResidentMemory** - Observed resident memory value
- **Data_VirtualMemory** - Observed virtual memory value

Office.Apple.MbuInstrument.VMOnShutdown

This event is collected for Office applications running under Apple platforms. The event is used to collect a

snapshot of the state of memory during application shutdown. This event helps us monitor our performance and provide performance improvements.

The following fields are collected:

- **Data_CollectionTime** - A timestamp from the moment in which the data was collected
- **Data_ResidentMemory** - Observed resident memory value
- **Data_VirtualMemory** - Observed virtual memory value

Office.Apple.MbuInstrument.VMOnStart

This event is collected for Office applications running under Apple platforms. The event is used to collect a snapshot of the state of memory during application start. This event helps us monitor our performance and provide performance improvements.

The following fields are collected:

- **Data_CollectionTime** - A timestamp from the moment in which the data was collected
- **Data_ResidentMemory** - Observed resident memory value
- **Data_VirtualMemory** - Observed virtual memory value

Office.Apple.MsoAppDelegate.BootPerf

This event is collected for Office applications running under Apple platforms. The event is used to collect time and memory consumed during boot by Office apps, as well as some details on the type of boot done. This event helps us monitor our performance and provide performance improvements.

The following fields are collected:

- **Data_AppLaunchDurationMicroSec** - The duration of the boot process
- **Data_AppLaunchFinishSystemTime** - A timestamp at a particular boot code marker
- **Data_AppLaunchStartSystemTime** - A timestamp at a particular boot code marker
- **Data_ResidentMemory** - A snapshot of the available resident memory during boot
- **Data_VirtualMemory** - A snapshot of the available virtual memory during boot

Office.Apple.UngracefulAppExitHangsInPreviousSession

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of our Office applications as well as for investigating causes of failures. We collect the number of times an application appeared to be unresponsive prior to hitting an ungraceful application exit.

The following fields are collected:

- **Data_HangsDetected** - The number of times the application appeared to become unresponsive prior to observing the ungraceful application exit.
- **Data_LastSessionId** - The identifier for the session in which the ungraceful application exit was observed.
- **Data_SessionBuildNumber** - The minor version of the application in which an ungraceful application exit was observed.
- **Data_SessionVersion** - The major version of the application in which an ungraceful application exit was observed.

Office.Apple.WhatsNewErrorAndWarning

This event is collected for Office applications running under Apple platforms. The event is used to monitor the

health of the What's New feature. This event denotes that an error/warning occurred while parsing What's New content, pointing to potential content authoring issues.

The following fields are collected:

- **Data_ContentKey** - A pointer to the section of the content that is likely to have caused the error.
- **Data_ErrorCode** - The observed error code (if available)
- **Data_ErrorDescription** - A description of the error (if available)
- **Data_EventID** - We collect a string representing the type of error observed.
- **Data_IncludesHTMLTag** - Whether the content contains rich html
- **Data_IncludesItemsTag** - Whether the content contains a hierarchy of items
- **Data_LengthOfRawData** - The size of the content
- **Data_RequestURL** - The URL from which the content was downloaded
- **Data_ServerLanguageTag** - The language the content was in.
- **Data_StatusCode** - The status of the error (if available)

Office.Extensibility.RichApiMethodInvocation

When customer uses an Office Add-in and calls Rich API for providing service, this event will be triggered. Used to measure the service reliability, performance, and usage for Rich API method invocation.

The following fields are collected:

- **Api** - Full name of the API
- **DispFlag** - A bit flag describing the type of method call (Ex: 0x1 = METHOD, 0x2 = PROPERTYGET, 0x4 = PROPERTYPUT, 0x8 = PROPERTYPUTREF)
- **DispId** - Dispatch ID for the method being called
- **HResult** - HResult for the method call
- **Latency** - Latency for the call, in microseconds
- **ReqId** - GUID for the batch request that this method belongs to
- **TypeId** - GUID for the interface on which this method being called

Office.Manageability.Service.ApplyPolicy

Critical telemetry to track failure\Success of applying cloud policy settings to registry. LastError tells why and where the Application of policy in registry failed.

The following fields are collected:

- **Data.ApplyLogMsg** - The exception msg if any while policy was being applied
- **Data.Cid** - dynamically generated correlation identifier sent to the service when the service call was made to fetch the cloud policy. Used to correlate which call caused an issue while applying the policies on the cloud.
- **Data.Last Error** - One of five string values (enumerators) to log which stage of policy application was being executed when the exception occurred

Office.OfficeMobile.PdfViewer.PdfFileOpenMeasurements (on Android)

This event is collected for the Office app for Android. It records when a file open operation takes place. We collect this data to ensure good performance for all file opens on the app.

The following fields are collected:

- **Data_Doc_ActivationFQDN** - Domain name of the Provider app for a file activation scenario (only first party app info is being logged).
- **Data_Doc_DownloadDurationms** - Time to download a PDF cloud file.
- **Data_Doc_Location** - Location where the file sits (Local, ODSP, iCloud, third-party files app, wopi
- **Data_Doc_OpenDurationms** - Time to open a PDF file in milliseconds.
- **Data_FetchReason** – Denotes how the file was fetched (manual, cached, not cached)
- **Doc_RenderDurationms** - Time to render a pdf file

Office.OfficeMobile.PdfViewer.PdfFileOpenMeasurements (on iOS)

This event is collected for the Office app for iOS. It records when a file open operation takes place. We collect this data to ensure good performance for all file opens on the app.

The following fields are collected:

- **Data_Doc_ActivationFQDN** - Domain name of the Provider app for a file activation scenario (only first party app info is being logged).
- **Data_Doc_CreateTelemetryReason** – Telemetry reason for PDF creation.(for example: Create from scan, using "picture to pdf" action, using "document to pdf" action, etc.)
- **Data_Doc_DownloadDurationms** - Time to download a PDF cloud file.
- **Data_Doc_DownloadEndTime** - Timestamp for end of download of a cloud file.
- **Data_Doc_DownloadStartTime** – Timestamp for start of download of a cloud file.
- **Data_Doc_FileOpSessionID** - Unique ID for a Document Session.
- **Data_Doc_Location** - Location where the file sits (Local, ODSP, iCloud, third-party files app, wopi
- **Data_Doc_OpenCompletionTime** - Timestamp for end of open operation of a PDF file.
- **Data_Doc_OpenDurationms** - Time to open a PDF file in milliseconds.
- **Data_Doc_OpenStartTime** - Timestamp for start of open operation of a PDF file.
- **Data_Doc_TelemetryReason** - Telemetry reason for the open event (for example: open from MRU or browse, File Activation, Protocol Activation, etc.).
- **Data_FetchReason** – Denotes how the file was fetched (manual, cached, not cached)
- **Doc_RenderDurationms** - Time to render a pdf file

Office.OneNote.Android.Sync.ProvisioningCompleted

[This event was previously named OneNote.Sync.ProvisioningCompleted.]

The critical signal used to ensure that after a user signs-into a OneNote Android App, notebooks are properly provisioned so that they can be easily accessed. This is used to ensure critical regression detection for OneNote app and service health

The following fields are collected:

- **AppSuspendedDuringEvent** - Returns Boolean to indicate if app was suspended during provisioning
- **NetworkConnection** - The type of network connectivity of the device in use

- **NetworkDataExchange** - Records the number of bytes exchanged during provisioning.
- **ServerType** - Returns the type of the server offering the service
- **TimeTakenInMilliSeconds** - Returns time taken to complete provisioning in millisecond

Office.OneNote.Android.Sync.ProvisioningError

The critical signal used to ensure that after a user signs-into a OneNote Android App, notebooks are properly provisioned so that they can be easily accessed. This is used to ensure critical regression detection for OneNote app and service health.

The following fields are collected:

- **AppSuspendedDuringEvent**: Returns boolean to indicate if app was suspended during provisioning
- **ErrorCode** – Returns the error code responsible for failure of provisioning
- **NetworkConnection**: The type of network connectivity of the device in use
- **NetworkDataExchange** - Records the number of bytes exchanged during provisioning.
- **ServerType**: Returns the type of the server offering the service
- **TimeTakenInMilliSeconds**: Returns time taken to complete provisioning in millisecond

Office.OneNote.Android.Sync.ProvisioningStarted

[This event was previously named OneNote.Sync.ProvisioningStarted.]

The critical signal used to ensure that after a user signs into a OneNote Android App, notebooks are properly provisioned so that they can be easily accessed. This is used to ensure critical regression detection for OneNote app and service health

The following fields are collected:

- **NetworkConnection** - The type of network connectivity of the device in use
- **ServerType** - Returns the type of the server offering the service

Office.OneNote.System.BootDialogs.SafeBootDialogPending

The critical signal used to track when we decide to show user a safe boot dialog on next boot because we have been crashing on boot multiple times continuously. This is used to ensure critical regression detection for OneNote app and service health. If users are seeing the safe boot dialog, then we have a critical boot crash bug and this info will help us know how many users are facing this issue and how many users boot the app again to actually see the safe boot dialog vs how many don't return.

The following fields are collected:

- None

Office.Outlook.Desktop.BootPerfMetrics

Collects time taken to boot Outlook. The boot time of Outlook is actively monitored to detect and diagnose regressions. It is also used to diagnose customer escalations and improve boot performance over time.

The following fields are collected:

- **AddinElapsedTotal** - Total time spent for loading add-ins
- **CredPromptCount** – Number of credential prompts shown
- **ElapsedTotal** - Total time spent for boot
- **IsLoggingEnabled** - Is logging enabled

- **ShowChooseProfileDlg** – Whether the Choose Profile Dialog was shown
- **ShowFirstRunDlg** - is outlook launched for the first time
- **ShowIMAPSrchfldWarningDlg** - Warnings in case we have an IMAP account with an ANSI PST
- **ShowNeedSupportDlg** - Boot failure triggered support dialog
- **ShowSafeModeDlg** - is the session opened in safe mode
- **ShowScanPstDlg** - Store repair check displayed error message

Office.Outlook.Mac.BootPerf

Collects time taken to boot Outlook. The boot time of Outlook is actively monitored to detect and diagnose regressions. It is also used to diagnose customer escalations and improve boot performance over time.

The following fields are collected:

- **MacOLKBootPerfDuration** - total time spent booting
- **MacOLKBootPerfID** - identifier for the time spent booting

Office.Outlook.Mac.PerformanceUnresponsive

Used to identify user impacting issues in Outlook that may manifest as degraded performance.

The following fields are collected:

- **Duration** - time elapsed of degraded performance
- **EventType** - type of event experiencing degraded performance

Office.Performance.Boot

Collected when an Office application is booted. Includes whether the boot was initiated by opening a file or launching via the Start menu, whether this was the first boot of the application, how much memory the application is using, and whether there was any blocking UI shown to the user. Used to measure how fast Office applications boot and how much memory they use when they start, to ensure there is an acceptable user experience.

The following fields are collected:

- **ActivationKind** - Whether the application was started by launching from the Start menu, by opening a file, or through OLE Automation.
- **BootToStart** - Whether the user has chosen to show the start screen when this application starts.
- **ChildProcessCount** – The number of child processes that have been launched by the application. (Windows only)
- **ColdBoot** - Whether this is the first time the Office application ran after a system restart or application binary had to be loaded from disk. (macOS/iOS only)
- **DeviceModel** - The model of the device. (macOS/iOS only)
- **DocAsyncOpenKind** - When opening a document, an enumeration indicating the type of asynchronous flow used.
- **DocLocation** - When opening a document, indicates which service provided the document (OneDrive, File Server, SharePoint, etc.).
- **DocSizeInBytes** - When opening a document, the file size in bytes.
- **DocSyncBackedType** - When opening a document, an indicator as to the type of document (local or service based)

- **DurationUntilMso20Initialization** - The duration in microseconds it took between when the Office process was initialized and mso20win32client.dll was loaded.
- **Embedding** – Whether the app was opened for OLE embedding.
- **FirstBoot** - Whether this was a first boot of the application.
- **FreeMemoryPercentage** – What percent of memory on the device is free. (Windows only)
- **HandleCount** – The number of operating system handles the process has opened. (Windows only)
- **HardFaultCount** – The number of hard page faults for the process. (Windows only)
- **InitializationDuration** - The duration in microseconds it took to first initialize the Office process.
- **InterruptionMessageId** - If the boot was interrupted by a dialog asking for user input, the ID of the dialog.
- **LegacyDuration** - The length of time the activity took to execute, measured using different starting and ending points than Activity.Duration.
- **OpenAsNew** – Whether the app was started by opening an existing document as the template for a new one.
- **OtherOperationCount** – The number of I/O operations performed, other than read and write operations. (Windows only)
- **OtherTransferCount** – The number of bytes transferred during operations other than read and write operations. (Windows only)
- **PageFaultCount** – The number of page faults for the process. (Windows only)
- **PrimaryDiskType** – Whether the primary storage device is a solid-state drive or a rotational drive and its rotation speed if applicable. (macOS/iOS only)
- **PrivateCommitUsageMB** – The Commit Charge (i.e., the amount of memory that the memory manager has committed for this process) in megabytes for this process. (Windows only)
- **PrivateCommitUsagePeakMB** - The largest Commit Charge in megabytes ever for this process. (Windows only)
- **PrivateWorkingSetMB** – The amount of memory in megabytes in the process's working set that isn't shared with other processes. (Windows only)
- **ProcessorCount** – The number of processors on the device. (macOS/iOS only)
- **ReadOperationCount** – The number of read operations performed. (Windows only)
- **ReadTransferCount** – The number of bytes read.
- **TotalPhysicalMemory** – The total amount of physical memory on the device. (macOS/iOS only)
- **TotalWorkingSetMB** - The amount of memory in megabytes in the process's working set.
- **VirtualSetMB** - The amount of memory in megabytes in the process's virtual set. (macOS/iOS only)
- **WorkingSetPeakMB** - The largest amount of memory in megabytes that was ever in the process's working set so far.
- **WriteOperationCount** – The number of write operations performed. (Windows only)
- **WriteTransferCount** – The number of bytes written. (Windows only)

Office.PowerPoint.PPT.Android.RehearseView

This event is denoting that user has stopped the rehearsal session. In combination with Office.PowerPoint.PPT.Android.RehearseView.StartSession this will be the first indicator of any crashes or errors that user faces.

The following fields are collected:

- **ConnectionCreationTime** - time taken to create service side connections.
- **CountDownAlertTime** – Time for which countdown alert was displayed.
- **CountdownInitTime–** Time between slideshow load completed and countdown started.
- **CritiqueSummary** - Summary of what all critiques user saw with their counts.
- **ExitEventCode** – Code to identify under which scenario user exit out of rehearse session, whether it was error scenario or successful exit.
- **FRETime** - Time between FRE screen started to display until user dismissed it.
- **MicrophonePermissionTime** - Time for which microphone permission alert was displayed until user select one of the options.
- **PauseRehearsingCount** – Count of how many times user clicked on pause rehearsal.
- **RehearsalInitTime** - Time taken by rehearsal to initialize.
- **ResumeRehearsingCount** – Count of how many times user clicked on resume rehearsal.
- **Sessionid** - This is speech frontdoor session ID. This is used to debug service logs.
- **SlideshowViewLoadTime** – Time taken by slideshow to load.

Office.PowerPoint.PPT.Android.RehearseView.RehearsalSummaryPage

Event triggered when summary page is loaded. This event helps us in capturing the performance of summary page. It tells how much time it takes for rehearsal summary service page to load on client. It is required to keep the feature performant.

The following fields are collected:

- **PayloadCreationTime** – This is the time taken in milliseconds to create payload.
- **PostUrlCallTime** – This is the time taken in milliseconds to send the post Url call.
- **RehearseSessionid** - This is speech frontdoor session ID. We can use this to debug service logs.
- **RequestPayloadSize** – This is the size of the request payload.
- **ResourcesLoadTime** – This is the time taken in milliseconds to load resources (js, css).
- **SummaryPageErrorReceived** – This is a Boolean value that indicates if summary page was received or error occurred.
- **SummaryPageHtmlLoadTime** – This is the time taken in milliseconds to load summarypageHtml.
- **SummaryPageLoadStartTime** – This is the time taken in milliseconds receive first response from the server.
- **SummaryPageLoadTime** – Time (in ms) taken to load summary page. This includes payload creation time
- **ThumbnailsCount** – This is the total number of thumbnails that will be part of summary page.

Office.PowerPoint.PPT.Android.RehearseView.StartSession

Event triggered when user clicks on start session. This event helps us in capturing how many users are using the feature of Presenter coach on Android. When combined with Office.PowerPoint.PPT.Android.RehearseView it will tell us how many users successfully completed the rehearsal session and how many couldn't. This is our first indicator of crashes or errors in the feature.

The following fields are collected:

- None

Office.PowerPoint.PPT.Shared.RehearseView.Errors

[This event was previously named Office.PowerPoint.PPT.Android.RehearseView.Errors]

Event triggered when any error occurs. This event will help us know what errors that user has faced and will help keep the Presenter Coach performant on mobile.

The following fields are collected:

- **Session ID** – rehearsal session ID
- **RehearsalEventCode** – rehearsal error code

Office.PowerPoint.Rehearsal.SessionMetrics

Event triggered when the speech session is stopped for Presenter Coach. This event helps us in capturing some metrics for a rehearsal session in Presenter Coach. It will help in maintaining a high quality of service for this feature.

The following fields are collected:

- **ActualRehearseBootTimeInMs** – This is the actual time taken for the connections to be created.
- **AdaptationTextSize** – This is the size of the text that is sent to service.
- **AuthDurationInMs** – This is the time taken in milliseconds for authentication (refresh the auth token).
- **AuthError** – This describes the authentication error that occurred (if at all).
- **AvgFragmentLatencyInMs** – This is the average round-trip time for network speech messages.
- **ConnectDurationInMs** – This is the time taken in milliseconds for the session to complete the connection.
- **FirstAudioDelayInMs** – This is the time taken in milliseconds for the first audio data to be received.
- **FRetriedOnOpenConnection** – This is a Boolean that indicates whether retry happens for openconnection or not.
- **InitMediaCaptureLayerDurationInMs** – This is the time taken in milliseconds to initialize the media/audio capture layer.
- **LocallyDroppedMessageCount** – This is the total number of messages dropped locally.
- **NumReconnectAttemptsDuringSession** – This indicates how many times the attempt was made to reconnect to speechservice.
- **NumTriesDuringEachReconnectAttempt** – This is an array that indicates number of tries that were done during each reconnect attempt.
- **OpenFrontDoorConnectionDurationInMs** – This is the time in milliseconds taken to open the connection to the FrontDoor service.
- **SendAdaptationTextDurationInMs** – This is the time taken in milliseconds to send the adaptation text to the service.

- **ServiceDroppedMessageCount** – This is the total number of messages dropped by the service.
- **SessionDurationInMs** – This is the time duration of entire session from when user clicked start to when user clicked stop.
- **SessionId** – This is the speech frontdoor session ID. We can use this to debug service logs.
- **SpeechClientResultEventsWithTimestamps** – This is an array of error codes received along with the timestamps, which can help in debugging.
- **SpeechHResultsWithTimestamps** – This is an array of error codes received along with the timestamps, which can help in debugging.
- **StartSpeechCaptureDurationInMs** – This is the time taken in milliseconds to start speech capture.
- **StartSpeechServiceDurationInMs** – This is an array of time taken to start speech session every time there is a reconnect, including first start speech session duration also.
- **TotalMessageCount** – This is the total number of audio messages sent to the service.
- **WebSocketConnectDurationInMs** – This is the time taken in milliseconds to complete the web socket connection.

Office.UX.OfficeInsider.CanShowOfficeInsiderSlab

Activity tracking whether the Office Insider slab can be shown to the user on the Account tab in the Office Backstage UI.

The following fields are collected:

- **Data_CanShow** - Indicates whether the Office Insider Slab can be shown to the user on the Account tab in the Office Backstage UI.
- **Data_Event** - Unused
- **Data_EventInfo** - Unused
- **Data_Reason** - Unused

Office.UX.OfficeInsider.RegisterCurrentInsider

Critical signal for tracking success or failure of registering users using Office Insider builds who weren't registered as Office Insiders before. Main scenario for this is current Office Insiders who joined Office Insider program before registration of Office Insiders was added.

The following fields are collected:

- **Data_RegisterInsider** - Status of Office Insider registration
- **Data_RegisterInsiderHr** - Result code for Office Insider registration
- **Data_RegistrationStateCurrent** - Current registration state
- **Data_RegistrationStateDesired** - Requested registration state

Office.UX.OfficeInsider.ShowOfficeInsiderDlg

Critical signal tracking user interaction with Join Office Insider dialog. It is used for identifying any issues in performing user-initiated changes such as joining or leaving Office Insider program and changing Office Insider level.

The following fields are collected:

- **Data_AcceptedContactMeNew** - Indicates if a user has accepted to be contacted by Microsoft upon joining Office Insider program

- **Data_InsiderLevel** - Insider Level on opening of "Join Office Insider" dialog
- **Data_InsiderLevelNew** - Insider level on closing of "Join Office Insider" dialog
- **Data_IsInternalUser** - Indicates whether the application runs under the credentials of a Microsoft corporate account.
- **Data_IsInternalUserInit** - Indicates whether the code could determine whether the application runs under the credentials of a Microsoft corporate account.
- **Data_OpenNewsletterWebpage** - Indicates whether Office Insider Newsletter Subscription link was triggered under condition that user has joined Office Insider program, Newsletter Subscription feature is enabled, and the user have not canceled opening of Office Insider Newsletter Subscription webpage.
- **Data_RegisterInsider** - Status of Office Insider registration
- **Data_RegisterInsiderHr** - Result code for Office Insider registration
- **Data_RegistrationStateCurrent** - Current registration state
- **Data_RegistrationStateDesired** - Requested registration state

Office.Visio.Shared.VisioFileRender

This event captures file render time. This event helps us keep file render performance in check.

The following fields are collected:

- **Data_AvgTime: integer** - Average time it took to render Visio drawing in a session
- **Data_CompositeSurfEnabled: bool** - true is composite rendering mode is enabled
- **Data_Count: integer** - Number of times Visio renders the drawing in a session
- **Data_FirstRenderTime: long** - duration to render file on first launch in millisecond
- **Data_MaxTime: integer** - Max time it took to render Visio drawing in a session

Office.Visio.VisioFileOpenReliability

This event collects File open perf data for Visio Dev16. This event is used to monitor performance of File open and associates it with file properties like file size for Visio Dev16. File properties enable us to debug and root cause issues faster.

The following fields are collected:

- **Data_CorrelationId: string** -Document correlation identifier
- **Data_DocIsEnterpriseProtected: bool** - true if document is protected with Windows information protection
- **Data_DocumentId: string** - GUID of file path
- **Data_DurationToCompleteInMilliseconds: double** - Duration to complete save as in millisecond
- **Data_DurationToCompleteInMillisecondsSquared: double** -squared value for DurationToCompleteInMilliseconds
- **Data_ErrorCode: integer** - Internal error code for file open failure
- **Data_Extension_Docs: string** - File extension of diagram opened
- **Data_FileIOBytesRead: int** - total bytes read while saving
- **Data_FileIOBytesReadSquared: int** - squared value of Data_FileIOBytesRead

- **Data_FileIOBytesWritten: int** - total bytes written while saving
- **Data_FileIOBytesWrittenSquared: int** - squared value of Data_FileIOBytesWritten
- **Data_FileName: binary** - Binary Hash of file name
- **Data_FileOpenDownloadDurationInMs: long** -duration to download file in milliseconds
- **Data_FileOpenEndDurationInMs: long:** -duration to open file in millisecond
- **Data_FileOpenTimeStamp: time:** -Time stamp when file started opening
- **Data_FilePathHash: binary** - GUID for file path
- **Data_FileSize: long** - Document size in bytes
- **Data_FileType: string** - File extension of diagram opened
- **Data_IsInternalFile: bool** - true if file is an internal file. for example, Stencil
- **Data_IsIRM: bool** - true if file is Information Right Protected
- **Data_IsReadOnly: bool** - true if the file is read only
- **Data_IsSuccess: bool** - true when file open was successful
- **Data_Location: string** - location of the file like Local, SharePoint, OneDrive, WopiConsumer, WopiBusiness, GenericThirdPartyConsumer
- **Data_NetworkIOBytesRead: int** - total network bytes read while saving
- **Data_NetworkIOBytesReadSquared: int** - squared value of Data_NetworkIOBytesRead
- **Data_NetworkIOBytesWritten: int** - total network bytes written while saving
- **Data_NetworkIOBytesWrittenSquared: int** - squared value of NetworkIOBytesWritten
- **Data_OpenLocation: integer** - Location of the file from which it was opened 0, Local, 1, Network, 2, SharePoint, 3 – Web
- **Data_Size_Docs: integer** - Document size in bytes
- **Data_Tag: string** - unique identifier to identify Save AS event
- **Data_WasSuccessful: bool** - true if open as was successful

OneNote.App.SafeBootDialogActionTaken, Office.OneNote.Android.SafeBootDialogActionTaken, Office.Android.EarlyTelemetry.SafeBootDialogActionTaken

The critical signal used to track user response when a safe boot dialog is shown. Safe boot dialog is shown when we were unable to launch repeatedly. This is used to ensure critical regression detection for OneNote app and service health. User sees when they encounter critical boot crash bug. This info will help track if the crash cause has been resolved and the user can launch the app successfully.

The following fields are collected:

- **DIALOG_ACTION** - Which dialog button did the user click on – Positive button or negative button

perf.event

Used for monitoring possible negative impact on performance of loading different parts of the app, for example to ensure when you first open the app, your inbox loads as quickly as possible.

The following fields are collected:

- **app_start_show_message_list** - that means there was a performance issue with the app start-up

causing your inbox message list to take a long time to load

- **average** - collects the number of reloads that happen on a conversation divided by the number of messages in that conversation.
- **event_type** - tells us the type of performance event that caused a performance issue to help us detect issues related to a specific type.
- **extra_params** - A developer can add additional parameters here to help give us more details about what could be causing this performance issue, that is, when did this action start and end, etc.
- **has_work_profile** - indicates whether the app is running under Android Work Profile or similar configuration, in order to correlate performance analysis to these environments.
- **profiling_summary** - provides information about the group of tasks, the number of tasks and the average time for those groups to help understand potential regressions in specific areas when loading the app
- **runtime_performance_monitoring_data** - provides the performance data (loading time, record count) when loading data in different parts of the app.
  - **average_cost_time_ns** - The average cost time measured in nanoseconds.
  - **cost_type** - Tells us whether this event is for measuring storage layer execution or total duration.
  - **hx_object_type** - Provides the detail programming object type of the measuring.
  - **is_main_thread** - Tells us whether this event only measures main thread execution time.
  - **record_count** - The number of records the underlying storage layer returns.
  - **scope_name** - Provides the name of UI page/components this event belongs to.
  - **total_cost_time_ns** - The total execution time measured in nanoseconds.
- **total_time_elapsed** - Tells us how long the performance event took to help us understand the severity of the performance issue

performance.record

This event collects performance metrics of the app. This allows us to detect and fix situations where the app memory usage and CPU usage becomes critically high or has other performance issues, which could cause your device to slow down.

The following fields are collected:

- **app_exit_metric** - Tell us the metrics about counts of different performance types of foreground and background app exits, to help us understand app exits unexpectedly with negative performance reasons.
- **average_suspended_memory** - Tells us the average amount of memory in use by the app when it's suspended so we have something to compare, to help us understand the negative performance impact.
- **category** - Tells us if the app is in the foreground or background at the time. Possible values include foreground and background.
- **cpu_usage** - Tells us how much CPU was used by the app so we have something to compare, to help us understand the negative performance impact
- **cumulative_CPU_time** - Tells us the total amount of CPU the app used with the measurement of duration of time, so we have something to compare, to help us understand the negative performance impact.
- **cumulative_GPU_time** - Tells us the total amount of GPU time used by the app, so we have something to compare, to help us understand the negative battery life impact.
- **is_watch_app_installed** - Tells us if the user is currently using an Apple Watch and whether it is

installed to help us understand the negative performance impact due to the Watch

- **is_watch_paired** - Tells us if the user is currently using an Apple Watch and whether it is paired with the device to help us understand the negative performance impact due to the Watch
- **is_watch_supported_and_active** - Tells us if the user is currently using an Apple Watch and whether it is active to help us understand the negative performance impact due to the Watch
- **memoAry_used_percentage** - Tells us what percentage of memory was used by the app so we have something to compare, to help us understand the negative performance impact
- **memory_used** - Tells us how much memory was used by the app so we have something to compare, to help us understand the negative performance impact
- **peak_memory_usage** - Tells us the largest amount of memory used by the app so we have something to compare, to help us understand the negative performance impact.
- **scroll_hitch_time_ratio** - Tells us the ratio of the time spent hitching while scrolling on UI, to help us understand the negative UI performance impact.

Application activity error subtype

Errors in functionality of a feature or user experience.

assertion

This event lets us detect when critical app errors occurred that would cause your app to crash or experience serious issues like causing you to see empty rows in your inbox.

The following fields are collected:

- **count** - Total number of items associated with the error; for example, number of calendars that have errors
- **has_hx** - Tells us the account is using our new sync service to help us detect issues caused by our sync service
- **host_name** - the name of the service host that was involved in the error to help us detect issues related to a specific host
- **host_type** - the type of host that was involved in the error to help us detect issues related to a specific host type
- **message** - custom message for the assertion that is used to diagnose the issue
- **origin** - the origin of the error in the code to help us detect issues related to a certain part of the code
- **stacktrace** - the stack trace where the assertion occurred to help us detect issues related to a certain part of the code
- **type** - the type of assertion error that occurred, for example, null_folder_name, compose_selected_null_account, to help us detect issues related to a certain part of the code

edit.contact.error

Allows us to detect and fix situations where errors were caused when you're trying to view or edit contacts through our app.

The following fields are collected:

- **errorType** - the type of error that occurred to help us diagnose the issue
- **field** - The contact field that the user was trying to edit to help us diagnose the issue
- **version** - The version of the contact card service we are using to help us diagnose the issue

error.report

This event lets us detect when critical app errors occurred so that we can prevent issues that could cause your app to crash or prevent you from reading email.

The following fields are collected:

- **client-request-id** - the client request identifier for the request that caused the error
- **date** - the datestamp of the request that caused the error
- **error** - the type of error, for example, get_mailbox_location_failed
- **error_body** - the body of the error message
- **is_x_mailbox_anchor_set** - whether the X-AnchorMailbox property was set on the request
- **reason** - the reason for the error, that is, an error message
- **request-id** - the server request identifier for the request that caused the error
- **source** - the source of the error within the OM infrastructure, typically one of 'BE' or 'FE'

Office.AirSpace.Backend.Win32.GraphicsDriverSoftHang

Helps Microsoft separate long video card driver hangs from short ones, which in turn helps make decisions about which video card drivers may be having problems. The user's video card driver has caused Office to hang, but the impact of the hang is not known yet

The following fields are collected:

- **Data_InDeviceCall** - The method called on the video card that produced the hang
- **Data_Timeout** - How long the hang lasted

Office.Android.ADALSignInUIPrompts

This event denotes that sign-in prompt came to the user, for school or work account. This event helps in understanding the health of signed in state of our apps and take appropriate actions, when we notice unexpected sign-in re-prompts.

The following fields are collected:

- **LastLoginDelta** - The time delta from last successful login.
- **PreviousIdentityCredProviderState** - Indicates the state of the account.
- **PreviousIdentityState** - Indicates state of the account, like session expired.
- **SignInResultCode** - Indicates the result code of sign-in prompt end.
- **UseCache** - Indicates if we force prompted the user to provide the password again.
- **UserType** - Indicates whether it is existing account or new account

Office.Android.AndroidAppDocsFileOperationEnds

Critical Docs Android Only (AppDocs) telemetry data for File New/Open/SaveAs end operations. This captures error codes for failures of these AppDocsOperations. Microsoft uses this to identify failures in various file operations and the exact layer at which the failure has occurred in Word, Excel, or PowerPoint.

The following fields are collected:

- **AccessMode** - enumeration value for the access mode for the file. Values- None, ReadOnly, ReadOnlyUpgradable, ReadWrite
- **BlockingUIShown** - Boolean to indicate if blocking UI was shown in the flow anywhere.

- **ContentUriAuthority** - the authority of the content URL from SAF
- **Correlation** - GUID for the correlation ID related to the operation
- **DocId** - the document ID generated by AppDocs
- **DocInstanceId** - DocInstanceId the document instance ID generated by AppDocs that is scoped to an operation instance on a document
- **DocIsEnterpriseProtected** - Boolean to indicate if document is protected.
- **DocUserId** - user ID from the MS auth layer
- **DocUserIdProvider** - enumeration that represents the user ID provider, 0 = Unkown, 1 = LiveId, 2 = OrgId, 3 = SSPI, 4 = ADAL
- **DurationInMs** - time in millisecond for the file operation to end
- **EndReason** - enumeration value for the end reason. Values - None, Success, Failure, Cancel
- **ErrorCode** - error code for the file operation
- **Extension** - extension of the file being opened.
- **FileSourceLocation** - enumeration value for file location. Possible values: None, Local, UncOrMappedNetworkDrive, SkyDrive, App, SharePoint, UnknownServer
- **FILETIME** - Time of the event
- **FirstBCSClientError_Info** - Error code information related to file conversions
- **HttpStatusCode** - http response code for web service request
- **InitalizationReason** - entry point for file open
- **K2FileIOHresult** - Hresult code for File open operation end
- **LastBCSClientError_TagId** - last error of BCS (binary conversion service) client
- **OfficeWebServiceApiStatusFlag** - status flag for the web service request
- **OpEndEventId** - tag that represents where the operation actually ended
- **OpFlags** - Document operation param flags used by AppDocs layer.
- **OpSeqNum** - A number that represents the sequencing of file operation related calls in AppDocs layer
- **OpType** - operation type enumeration. Values: "None", "CreateDocument", "OpenDocument", "CopyDocument", "CloseDocument", "SaveDocument", "OpenVersion", "CloseVersion"
- **PreFetchState** - enumeration for prefetch state of templates for new file create operations.
- **ProviderApp** - the package name of the app from which file is opened
- **ScopeInstanceId** - Scope Instance ID used to join data context to activities
- **Size** - file size
- **State** - enumeration value for the state of the file. Values: None, Creating, Created, CreateFailed, Opening, Opened, OpenFailed, Copying, Copied, CopyFailed, Closing, Closed, CloseFail
- **TemplateName** - the binary template name of the document template from the template service, for example, TF10002009.dotx
- **UriScheme** - scheme of the URL

Office.Android.AndroidAuthError

This event denotes core authentication failures during silent token refresh, loading sign in page from service and so on. This event helps in understanding the health of signed in state of our apps, sign in attempts that are made, and take appropriate actions, when we notice unexpected failures.

The following fields are collected:

- **ADALErrorCode** - Indicates error code while showing sign-in prompt or silent token fetch attempt for work account.
- **ADALRawErrorCode** - Indicates raw error code while showing sign-in prompt or silent token fetch attempt for work account.
- **ErrorGroup** - Indicates the type of account like personal account or work account or on-premise work account.
- **IDCRLErrorCode** - Indicates error code while showing sign-in prompt for personal account.
- **IDCRLRawErrorCode** - Indicates raw error code while showing sign-in prompt for personal account.
- **LiveOAuthErrorCode** - Indicates error code during silent token refresh attempt for personal account.
- **LiveOAuthRawErrorCode** - Indicates raw error code during silent token refresh attempt for personal account.
- **NTLMErrorCode** - Indicates error code while showing sign-in prompt for on-premise work account.

Office.Android.AndroidFileAsyncSaveStatus

Captures File async save status data and various error codes from different components. Microsoft uses this data to analyze if there is any user data loss in the app during saving of files in Word, Excel, or PowerPoint.

The following fields are collected:

- **FileExtension** - file extension
- **FileIOSaveHResult** - HResult for file save operation
- **FileIOSaveIsCopy** - Boolean to indicate if operation is saving a copy.
- **FileSize** - size of file
- **FileSourceLocation** - enumeration for file source location. Values: None, Local, UncOrMappedNetworkDrive, SkyDrive, App, SharePoint, UnknownServer

Office.Android.AndroidFileOpenReliability

This captures File open status data and various error codes to identify what file open failures are expected versus unexpected and which part of the code is reporting them. Microsoft uses this data to analyze the reasons for file open failures and calculate critical metric like file open success rate in Word, Excel, or PowerPoint.

The following fields are collected:

- **AccessMode** - Access mode enumeration
- **AppDocsFileOpenErrorCode** - AppDocs error code for file open failure
- **ContentUriAuthority** - authority of the content URL from SAF
- **DownloadCsiError** - download error message from CSI
- **FileExtension** - file extension
- **FileOpenEndErrorCode** - Error code for file open failure

- **FileOpenStatus** - File open status enumeration
- **FileSize** - file size
- **FileSourceLocation** - File Location enumeration
- **FirstBCSClientError_Info** - last error of BCS (binary conversion service) client
- **IfWordFileOpencanceled** - if file open was canceled by user in Word
- **InitializationReason** - enumeration for the entry point for file open
- **IsAutoSaveDisabled** - Is auto save disabled during file open
- **IsFileEmpty** - Boolean to indicate if file is empty
- **K2FileIOHresult** - Hresult for File operation end
- **OpenCsiError** - file open error message in the CSI layer
- **OpEndEventId** - tag where the operation actually ended
- **PPTHresult** - Hresult in PPT
- **PPTIsExpectedError** - PPT Error classification for file open expected/unexpected failure
- **PPTTag** - error tag in PPT
- **ProviderApp** - the package name of the app from which file is opened
- **ProviderFileSize** - file size captured while opening file via file activation
- **State** - File open state enumeration
- **UriScheme** - Scheme of the URL
- **WordErrortag** - error tag in Word
- **WordFileCorruptionReason** - Reason for corruption due to which word file can fail in opening
- **WordFileOpenErrorCode** - Word-specific file open error code.
- **WordFileTypeFromDod** - File type determined by word based on actual file format
- **WordFileTypeFromExtension** - File type determined by word based on file extension

Office.Android.AndroidFileSaveStatus

Critical to capture File save status data and various error codes from different components. Microsoft uses this data to analyze if there is any user data loss in the app during saving of files in Word, Excel, or PowerPoint.

The following fields are collected:

- **AccessMode** - Values** - None, ReadOnly, ReadOnlyUpgradable, ReadWrite.
- **AppDocsEndReason** - enumeration for File save Appdoc EndReason. Values: None, Success, Failure, Cancel.
- **AppDocsErrorCode** - Final Error Code for file save failure
- **AppDocsTriggeringSaveDetails** - field to indicate if AppDocs is triggering the save
- **DocInstanceId** - DocInstanceId the document instance ID generated by AppDocs that is scoped to an operation instance on a document
- **ExcelFileSaveResult** - Excel-specific HResult

- **FileExtension** - Extension of file.
- **FileIOSaveErrorCode** - Error code in FileIO
- **FileIOSaveHResult** - Hresult in FileIO
- **FileIOSaveIsCopy** - Boolean to indicate if this is a copy operation
- **FileSize** - Size of file
- **FileSourceLocation** - File location enumeration. Values: None, Local, UncOrMappedNetworkDrive, SkyDrive, App, SharePoint, UnknownServer
- **OpFlags** - Operation flags for save
- **PPTFileSaveFailHresult** - PPT hresult for save failure
- **PPTFileSaveFailTag** - PPT tag for save failure
- **State** - File Open state enumeration.
- **Values** - None, Creating, Created, CreateFailed, Opening, Opened, OpenFailed, Copying, Copied, CopyFailed, Closing, Closed, CloseFail
- **WordFileCopyErrorTrackbackTag** - trackback tag for failure is CopyDocument stage in Word
- **WordFileSaveCancelReason** - trackback tag for cancels in word
- **WordFileSaveEid** - Word-specific error code
- **WordFileSaveErrorTrackbackTag** - trackback tag for save failures
- **WordFileSaveOpResult** - enumeration for result status 0 if succeeded, 1 if failed, 2 if canceled
- **WordFileSaveSuccess** - enumeration for Word-specific details for file save operation success.

Office.Android.AndroidOfficeActivationLatency

Critical data to collect end-to-end file open time for all file opens in Windows, Excel, PowerPoint apps. This is used by Microsoft to find out the metric for file open performance of our apps

The following fields are collected:

- **AppBootingOccured** - Boolean to check if app boot is complete
- **ApplicationBootTime** - time required during a specific phase of app boot
- **AppSuspendedDuringBoot** - Boolean to check if app was suspended during boot
- **BlockingUIShownDuringFileOpen** - Boolean to indicate if there was any blocking dialog during the file open operation
- **CachedInfoAvailable** - Boolean to look for cached info specific to file open operation
- **DocumentRecoveryInvoked** - Boolean to indicate if there was a document pending for recovery
- **EndToEndActivationTime** - time taken to render the file for files opened from outside the app
- **EndToEndFileOpenTime** - time taken to render the file for files opened from inside the app
- **FileOpenPhaseDurationInMs** - file open operation time consumed by specific phase
- **FileSourceLocation** - enumeration value for File location such as None, Local, UncOrMappedNetworkDrive, SkyDrive, App, SharePoint, UnknownServer
- **InitalizationReason** - entry point for file open

- **InitialBootPhaseTime** - time required during a specific phase of app boot
- **IsThisFirstLaunch** - Boolean to indicate if this is the first launch of the app
- **MinimumLibraryLoadPhaseTime** - time required during a specific phase of app boot
- **MinimumLibraryLoadPhaseTime** - time required during a specific phase of app boot
- **MinimumLibraryLoadPhaseTime** - time required during a specific phase of app boot
- **PostAppInitTimeInMs** - time required during a specific phase of app boot
- **PPTRenderPhase** - time related to specific phase in PPT rendering
- **PreAppInitTimeInMs** - time required during a specific phase of app boot
- **ProviderApp** - the package name of the app from which file is opened
- **TelemetryReason** - similar to InitialisationReason, but more detailed enumeration value regarding the entry point for File open.
- **UserDialogInterruptionDuringBoot** - Boolean to indicate if there was any blocking dialog during the boot
- **XLRenderPhase** - time related to specific phase in Excel rendering

Office.Android.AppDocsFileOperationEnds

Critical Docs Android Only (AppDocs) telemetry data for File New/Open/SaveAs end operations. This captures error codes for failures of these AppDocsOperations. Microsoft uses this to identify failures in various file operations and the exact layer at which the failure has occurred in Word, Excel, or PowerPoint.

The following fields are collected:

- **AccessMode** - enumeration value for the access mode for the file. Values: None, ReadOnly, ReadOnlyUpgradable, ReadWrite
- **BlockingUIShown** - Boolean to indicate if blocking UI was shown in the flow anywhere.
- **ContentUriAuthority** - the authority of the content URL from SAF
- **Correlation** - GUID for the correlation ID related to the operation
- **DocId** - the document ID generated by AppDocs
- **DocInstanceId** - DocInstanceId the document instance ID generated by AppDocs that is scoped to an operation instance on a document
- **DocIsEnterpriseProtected** - Boolean to indicate if document is protected.
- **DocUserId** - user ID from the MS auth layer
- **DocUserIdProvider** - enumeration that represents the user ID provider, 0 = Unkown, 1 = LiveId, 2 = OrgId, 3 = SSPI, 4 = ADAL
- **DurationInMs** - time in milliseconds for the file operation to end
- **EndReason** - enumeration value for the end reason. Values: None, Success, Failure, Cancel
- **ErrorCode** - error code for the file operation
- **Extension** - first four characters of the extension of the file being opened.
- **FileSourceLocation** - enumeration value for file location. Possible values: None, Local, UncOrMappedNetworkDrive, SkyDrive, App, SharePoint, UnknownServer

- **FILETIME** - Time of the event
- **FirstBCSClientError_Info** - Error code information related to file conversions
- **HttpStatusCode** - HTTP response code for web service request
- **InitalizationReason** - entry point for file open
- **K2FileIOHresult** - Hresult code for File open operation end
- **LastBCSClientError_TagId** - last error of BCS (binary conversion service) client
- **OfficeWebServiceApiStatusFlag** - status flag for the web service request
- **OpEndEventId** - tag that represents where the operation actually ended
- **OpFlags** - Document operation param flags used by AppDocs layer.
- **OpSeqNum** - A number that represents the sequencing of file operation related calls in AppDocs layer
- **OpType** - operation type enumeration. Values: "None", "CreateDocument", "OpenDocument", "CopyDocument", "CloseDocument", "SaveDocument", "OpenVersion", "CloseVersion"
- **PreFetchState** - enumeration for prefetch state of templates for new file create operations.
- **ProviderApp** - the package name of the app from which file is opened
- **ScopeInstanceId** - Scope Instance ID used to join data context to activities
- **Size** - file size
- **State** - enumeration value for the state of the file. Values: None, Creating, Created, CreateFailed, Opening, Opened, OpenFailed, Copying, Copied, CopyFailed, Closing, Closed, CloseFail
- **TemplateName** - the binary template name of the document template from the template service, for example, TF10002009.dotx
- **UriScheme** - scheme of the URL

Office.Android.AuthACEErrors

This event uses the Microsoft Account (MSA) to determine which user is attempting to sign in to the app and during that which telemetry in discussion is getting triggered as part of an unsuccessful attempt.

This event helps with MSA sign-in error distribution analysis, which helps in understanding reasons behind unsuccessful MSA sign-in flow end.

The following fields are collected:

- **ExceptionsName** - indicates exception classes with regard to exception tags that occur during Microsoft account sign-in flow.
- **ExceptionsTag** - indicates which inflow exceptions present in union are occurring for MSA-sign in flow.
- **IDCRLACEErrorCode** - Gives error code occurring during MSA sign-in flow. Different error codes mentioned at %SRCROOT%\identity\coreapi\public\IdentityData.h
- **IDCRLAuthenticationStatusErrorCode** - Indicates error codes for invalid status of Authentication result coming from Microsoft Account (MSA).
- **IDCRLUserInteractionMissingError** - Indicates if Microsoft Account (MSA) sign-in flow invoked with showUI flag as false causing the hit.

Office.Android.BCS.Errors

Binary conversion Errors telemetry for Print and Share as PDF. Microsoft uses this to identify failure points during BCS conversions in Word, Excel, or PowerPoint.

The following fields are collected:

- **DocumentFileSize** - File size.
- **FileExtension** - First four characters of the extension of the file.
- **IsFileDirty** - Boolean to indicate if there were unsaved changes in the file.
- **Location** - File location enumeration. Values: OneDrive, SharePoint, Dropbox, Others
- **PDFConversionError** - Tag at which error occurs for PDF conversion
- **PdfConversionErrorCode** - PDF conversion error code
- **PdfConversionHRStatus** - PDF conversion status code
- **PdfConversionResult** - PDF Conversion result enumeration. Values: "Success", "Failed" and "Canceled"
- **PdfFileSize** - Size of the PDF

Office.Android.ClientSideIAP

Critical Error telemetry for Database Failure while file browsing and adds places. Microsoft uses this to identify DB corruption issues in the apps which might hinder user to add places or browse through them from within the app in Word, Excel, or PowerPoint.

The following fields are collected:

- **ClientTransactionId** - GUID passed to DSC for a specific Redemption request.
- **CollectionTime** - time of subscription purchase completion
- **CountryCode** - Client country code that is sent to DSC for client redemption request
- **GoPremiumEntryPoint** - entry point for triggering purchase
- **IsActivateExistingSubscription** - Boolean to indicate if there was an existing subscription that was activated
- **IsErrorRetriable** - Boolean to indicate if redemption can be retried
- **IsPreviousPurchase** - Boolean to indicate if activation occurred with a previous purchase of subscription
- **IsProvisioningTriggeredByRetry** - Boolean to indicate if retry was involved
- **LanguageCode** - Client language code that is sent to DSC for client redemption request
- **ProductIdentifier** - SKU name that the client is trying to purchase
- **ProvisioningHttpStatusCode** - Provisioning http status code
- **ProvisioningStatusCode** - Provisioning status code
- **PurchaseOrderId** - Purchase order identifier from Google/Samsung store
- **RedemptionTaskHR** - HResult for redemption task of subscription
- **SubscriptionProvisioningSucceeded** - Boolean for subscription provisioning result success
- **SubscriptionPurchaseHR** - Hresult for subscription purchase task
- **SubscriptionType** - enumeration for subscription type or SKUs.

- **TCID** - Icon click the triggers the subscription flow

Office.Android.DBFailureCause

Critical Error telemetry for Database Failure while file browsing and adds places. Microsoft uses this to identify DB corruption issues in the apps which might hinder user to add places or browse through them from within the app in Word, Excel, or PowerPoint.

The following fields are collected:

- **ErrorAt** - Tag Value: Information about the place where the Failure happened
- **ExceptionErrorMessage** - verbose error message

Office.Android.EarlyTelemetry.ExpansionFilesErrors

Android Package Kit (APK) expansion files for the Office mobile app are supplementary resource files that Android app developers can publish along with their app. To make our Expansion files download mechanism more reliable, we are logging the cause of errors that occur either in downloading the expansion files or while reading the downloaded expansion files.

The following fields are collected:

- **Data_ClassName** - Text representing the source code file name where there is an error.
- **Data_ErrorMessage** - Text representing the operation that has failed.
- **Data_ExceptionMessage** - An optional text field representing the cause of the exception.
- **Data_ExceptionType** - An optional text field representing the name of the exception thrown from source code.
- **Data_MethodName** - Text representing the method name in source code where there is an error.

Office.Android.EarlyTelemetry.ExtractionError

To reduce the size of Office Android apps, we apply compression to the resources in the final package. At run time, we first extract these resources before using them. Sometimes there are unexpected errors while performing extractions, which leads to app crashes.

Through this event we are collecting some diagnostic information related to extraction, like name of the resource being extracted, path where its extracted etc., free disk space available etc. This data is collected only when there are extraction errors.

We use this data to understand the cause of extraction failures, and to improve the user experience of our apps.

The following fields are collected:

- **Data_ArchiveName** - Name of the resource that is being extracted.
- **Data_ArchivePath** - Path where the resource is temporarily cached.
- **Data_ArchiveSizeKB** - Size of the resource that is being extracted.
- **Data_ClassName** - File name in source code where the error is encountered.
- **Data_ErrorDetail** - Text describing more details about the cause of the error, like the error code etc.
- **Data_ErrorMessage** - Text describing the type of error encountered during extraction.
- **Data_ExtractionDestinationPath** - Path where the resource is to be saved after extraction.
- **Data_FreeDiskSpaceMB** - The amount of free disk space available on the device measured in Mega Bytes.
- **Data_ItemToExtract** - Name of the resource that is being extracted.

- **Data_MethodName** - Method name in source code where the error is encountered.

Office.Android.EarlyTelemetry.RegistryErrors

This event captures any errors faced during Android registry access. This event data helps us in understanding the user errors and making the registry feature more robust.

The following fields are collected:

- **App** – The application process sending the event.
- **AppVersionLong** – The application version.
- **Data_StackTrace** – The stacktrace of the error.

Office.Android.EarlyTelemetry.SharedLibraryLoadersearchAndloadLibraryError

We log this event in case there are errors while loading shared libraries. There can be library loading errors for two reasons 1) Installed Apk is incompatible with the device. 2) The library that we are trying to load may be corrupt, because of errors in extracting it due to low disk space, or low memory.

The following fields are collected:

- **Data_ExceptionMessage** - Exception message thrown by Android API System.loadlibrary
- **Data_FreeSpaceInMB** - Free space available on device
- **Data_nickName** - Name of the library that couldn't be loaded.

Office.Android.Intune.IntuneJavaCopyFailedAttempts

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to save local copy of Intune protected Cloud documents. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- **Data_FileCreationFailedErrorCode** - Error code associated with the flow

Office.Android.Intune.IntuneJavaExceptionADALTokenForMAM

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to obtain the ADAL token for Intune resources. Microsoft uses this data to identify errors during Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- **Data_ErrorCode** - Error code associated with the flow

Office.Android.Intune.IntuneJavaExceptionAppPolicy

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to fetching policies for an identity for the current process. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionAppPolicyForContext

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to fetching policies for an identity for the current activity. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionAuthenticationCallback

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to registering for authentication callbacks for managed accounts. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionGetAccountStateSync

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to managed account. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionGetIsSaveToLocationAllowed

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to fetch the policy related to save to local. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionGetPolicyForIdentity

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to fetching policies for an identity. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionGetProtectionInfoFromDescriptor

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to Protection Info. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionGetProtectionInfoFromPath

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to Protection Info. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionGetUIPolicyIdentity

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to fetching UI policies for a managed account. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionIsIdentityManaged

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to identifying if an account is managed. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account.

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionNullEnrollmentManager

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to registration of components for callback. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionProtect

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to protecting a managed document. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account.

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionProtectFromDescriptorIfRequired

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to protecting a managed document. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionRegisterAccountSync

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to registering account Intune Management. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionSetUIPolicyIdentitySync

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to setting policies for a managed account. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionUnregisterAccountSync

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to remote wipe scenarios for Intune Management. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- None

Office.Android.Intune.IntuneJavaExceptionUpdateToken

Critical Error telemetry to track failures for individual Intune APIs; This telemetry gets logged in case of errors to call Intune APIs related to update authorization token for a managed account. Microsoft uses this data to identify errors during and after Intune enrollment within the app, post signing into the app with a work account

The following fields are collected:

- None

Office.Android.LicenseActivationFailure

Critical Error telemetry to track failures to activate licenses for Office 365 accounts in Word, Excel, or PowerPoint. Microsoft uses this to analyze failures to activate a purchased Office 365 license.

The following fields are collected:

- **EntryPoint** - Entrypoint enumeration for triggering the license activation flow
- **HResult** - error code for the failure
- **IsGallatin** - Boolean to check if it is a Gallatin account
- **MessageCode** - enumeration to indicate the point of failure for activation
- **PreviousEntryPoint** - Entrypoint enumeration for triggering the license activation flow
- **StateAfterActivation** - enumeration to indicate licensing state of the app before the activation flow started
- **StateBeforeActivation** - enumeration to indicate licensing state of the app before the activation flow started
- **UserAccountType** - enumeration to indicate whether personal account or enterprise account.

Office.Android.MSASignInUIPrompts

This event denotes that sign-in prompt came to the user, for personal account. This event helps in understanding the health of signed in state of our apps and take appropriate actions, when we notice unexpected sign-in re-prompts.

The following fields are collected:

- **ExternalCacheRefreshError** - Error code of token refresh attempt, before showing sign-in prompt.
- **LastLoginDelta** - The time delta from last successful login.
- **MSAserverUAID** - Correlation ID with service telemetry data.
- **PreviousIdentityState** - Indicates state of the account, like session expired.
- **SignInResultCode** - Indicates the result code of sign-in prompt end.
- **UseCache** - Indicates if we force prompted the user to provide the password again.
- **UserType** - Indicates whether it is existing account or new account
- **WasIdentitySignedOut** - Indicates if account was in signed out state.

Office.Apple.Licensing.Mac.DRActivationFailures

This event is collected for Office applications running under Apple platforms. The event is used for capturing digital river activation failures (the event logs the key and product that was used to activate, and the error code

received). This event is used for detecting and helping troubleshoot activation failures (Digital River issues).

The following fields are collected:

- **Data_DigitalRiverID** - Digital River product ID that maps to this Office product SKY
- **Data_Error** - A string representing an activation error code.
- **Data_ProductKey** - Product key that was attempted to be activated
- **Data_ProductKeyHash** - An encoded product key being activated

Office.Apple.Licensing.Mac.GetMachineStatusErrors

This event is collected for Office applications running under Apple platforms. The event collects the error code returned while periodically checking the validity of a subscription license. The error code can signify server unavailability but also license expiration, machine count limit, invalid hardware ID, etc. This event is used to monitor the health of the Office Licensing Service but also to investigate issues related to subscription machine management.

The following fields are collected:

- **Data_Error** - We collect a string representing an error code.

Office.Extensibility.Sandbox.ODPErrorNotification

Tracks the various error notifications received from the sandbox. Used to detect the error scenarios in sandbox and there by fixing it, to improve productivity of the user

The following fields are collected:

- **AppId** - ID of the App
- **AppUrl** - scrubbed app URL
- **Result** - result error code

Office.FirstRun.Apple.MacONIOLKFirstRunStarted

This event is collected for Office applications running under Apple platforms. The event lets us know a user has entered first run experience. We use this event to figure out if the First-Run Experience (FRE) was started successfully.

The following fields are collected:

- **Data_FirstRunCollectionTime** - A timestamp registering the time at which the flow was started.

Office.Graphics.ARCExceptions

This exception reporting information is important for assessing the overall health of the graphics stack, as well as identifying parts of the code where failures are occurring at high frequency, in order to prioritize investigation. This exception reporting information is important for assessing the overall health of the graphics stack, and identifying parts of the code where failures are occurring at high frequency. This helps an engineer to determine which rendering failures are impacting the most users, enabling us to prioritize our investigations toward fixing issues that will have the greatest user benefit.

The following fields are collected:

- **Data_HResult** - The error code returned from failure
- **Data_TagCount** - The count of each failure that occurred
- **Data_TagID** - The identifier of the failure that occurred

Office.OfficeMobile.PersonalizedCampaigning.Errors

To raise awareness about the features of Office mobile that users have not yet explored, Office mobile integrates

with IRIS to support in-app and push notifications. In case of in-app notifications, it captures errors that happen while pulling or displaying notification and when user interactions with the notification as well as providing feedback to IRIS server. In case of push notifications, it captures errors that happen while displaying notification, and when user interacts with the notification.

The following fields are collected:

- **Class** - Name of the class where error occurred
- **CreativeId** - The ID of notification which uniquely identifies the notification and its content.
- **ErrorDetails** - Details on the error
- **ErrorMessage** - Error message.
- **ErrorReason** - The underlying reason for the error
- **Method** - Name of the function where error occurred.
- **RequestParams** - Request parameters used when contacting the IRIS server to pull the notification.
- **SurfaceId** - ID of the surface where the notification will be shown.

Office.Outlook.Desktop.Calendar.AcceptCalShareNavigateToSharedFolder.Error

Collects information when any failure occurs when while Navigation to shared Calendar. This data is used to monitor the health of the calendar sharing API and Outlooks interaction with shared calendars.

The following fields are collected:

- **FailedCaseHResult** - The error code returned from Failure

Office.Outlook.Desktop.EDP.EDPOpenStoreFailure

Success or failure to open the Enterprise Data Protection protected mail store based on result of Windows API call to get the key to decrypt the store. We use this diagnose one of the top Enterprise Data Protection issues, which can prevent Outlook from booting. Primary cause of failure is Outlooks interaction with Windows APIs used to decrypt the store key.

The following fields are collected:

- **HVA Activity** - with custom data fields
- **IsFlightOn** – Indicates whether the EDPDecryption Flight is enabled

Office.Outlook.Desktop.NdbCorruptionResult

Office.Outlook.Desktop.NdbCorruptionResult and Office.Outlook.Desktop.NDBCorruptStore.Warning are collected when we detect corruption in a user's PST/OST. When we detect corruption, Microsoft collects the format of the database, the place where detected it, and a small amount of context about the corruption. OST/PST corruption prevents users from accessing their emails. We actively monitor this data for anomalous activity. We aim to investigate and diagnose issues to limit loss of customer data.

The following fields are collected:

- **0** - The Process name that reported corruption
- **1** - Bool indicating if the user chooses new file or not
- **2** - the number of other processes that have the database open

Office.Outlook.Desktop.NDBCorruptStore.Warning

Office.Outlook.Desktop.NdbCorruptionResult and Office.Outlook.Desktop.NDBCorruptStore.Warning are collected when we detect corruption in a user's PST/OST. When we detect corruption, Microsoft collects the format of the database, the place where detected it, and a small amount of context about the corruption.

OST/PST corruption prevents users from accessing their emails. We actively monitor this data for anomalous activity. We aim to investigate and diagnose issues to limit loss of customer data.

The following fields are collected:

- **CollectionTime** - collection time
- **Context** - Corrupt Store Context, where corruption was detected
- **CreatedWithVersion** – (Optional) field with version of store
- **Details** – Details about the crash
- **NdbType** - Store Type, can be 0 = NdbUndefined, 1 = NdbSmall, 2 = NdbLarge, 3 = NdbTardis
- **ProcessName** - Process Name causing the store to get corrupted
- **PstVersion** - Version of the MSPST32.DLL
- **Version** - Version of store file format

Office.Outlook.Desktop.OutlookCalendarUsageErr.MeetRcpt.ForwardActions.Rule.O16

Collects success and failure of the Forward, Forward as Attachment, and Forward as iCalendar action for Single, Recurring, and Exceptional Meeting Responses in the Mail, Calendar, and Inspector Outlook view. The failure rate of the Forward, Forward as Attachment, and Forward as iCalendar actions are actively monitored for anomalies. Anomalous statistics indicate a failure Outlooks ability to conduct core calendar operations. This data is also used to diagnose other Calendar-related issues that may be detected.

The following fields are collected:

- **CountExceptionForward** - Count of the forwarded Meetings Exceptions
- **CountExceptionForwardAsiCal** - Count of the forwarded Meetings Exceptions as an iCal
- **CountExceptionForwardInSplit** - Count of the forwarded Meetings Exceptions from the Split Menu in Ribbon
- **CountExceptionForwardWithAttach** - Count of the forwarded Meetings Exceptions as an Attachment
- **CountRecurringForward** - Count of the forwarded Recurring Meetings
- **CountRecurringForwardAsiCal** - Count of the forwarded Recurring Meetings as an iCal
- **CountRecurringForwardInSplit** - Count of the forwarded Recurring Meetings from the Split Menu in Ribbon
- **CountRecurringForwardWithAttach** - Count of the forwarded Recurring Meetings as an Attachment
- **CountSingleForward** - Count of the forwarded Single Meetings
- **CountSingleForwardAsiCal** - Count of the forwarded Single Meetings as an iCal
- **CountSingleForwardInSplit** - Count of the forwarded Single Meetings from the Split Menu in Ribbon
- **CountSingleForwardWithAttach** - Count of the forwarded Single Meetings as an Attachment
- **HResult** - ErrorCode
- **OlkViewName** - Indicates Mail, Calendar, or Inspector View

Office.Outlook.Desktop.OutlookCalendarUsageErr.MeetRcpt.ReplyActions.Rule.O16

Collects success and failure of the Reply, Reply All, Reply With IM, and Reply All with IM action for Single, Recurring, and Exception Meeting Responses in the Mail, Calendar, and Inspector Outlook view. The failure rate of the Reply, Reply All, Reply With IM, and Reply All with IM actions are actively monitored for anomalies.

Anomalous statistics indicate a failure Outlooks ability to conduct core calendar operations. This data is also used to diagnose other Calendar-related issues that may be detected.

The following fields are collected:

- **CountExceptionReply** - Count of the Meetings Reply on exceptions
- **CountExceptionReplyAll** - Count of the Meetings ReplyAll on exceptions
- **CountExceptionReplyAllWithIM** - Count of the Meetings ReplyAll with IM on exceptions
- **CountExceptionReplyWithIM** - Count of the Meetings Reply with IM on exceptions
- **CountRecurringReply** - Count of the Recurring Meetings Reply
- **CountRecurringReplyAll** - Count of the Recurring Meetings ReplyAll
- **CountRecurringReplyAllWithIM** - Count of the Recurring Meetings ReplyAll with IM
- **CountRecurringReplyWithIM** - Count of the Recurring Meetings Reply with IM
- **CountSingleReply** - Count of the Single Meetings Reply
- **CountSingleReplyAll**- Count of the Single Meetings ReplyAll
- **CountSingleReplyAllWithIM** - Count of the Single Meetings ReplyAll with IM
- **CountSingleReplyWithIM** - Count of the Single Meetings Reply with IM
- **HResult** - ErrorCode
- **OlkViewName** - Indicates Mail, Calendar, or Inspector View

Office.Outlook.Desktop.OutlookPrivsDlgSingleUser.LoadFail

This rule collects Calendar Sharing errors when adding a new user (of type EX or SMTP) from the Address book. This data is used to diagnose and resolve issues detected in the Calendar Sharing dialog

The following fields are collected:

- **CountAccountWizardEnd** - How many times the legacy wizard dialog ended
- **CountCreatePIMAccount** - How many times user created a PIM Profile

Office.Outlook.Mac.MacOLKAsserts

Used to identify user impacting issues in Outlook that may manifest as crashes or degraded functionality.

The following fields are collected:

- **Category** - type of assert
- **CollectionTime** - time when assert is collected

Office.Outlook.Mac.MacOLKErrors

Used to identify user impacting issues in Outlook that may manifest as crashes or degraded functionality.

The following fields are collected:

- **Category** - type of error
- **CollectionTime** - time when error is collected
- **ThreadId** - identifier for the thread

Office.System.SystemHealthAsserts

The errors this event identifies help us understand when the customer experience is degrading. Many of these

ShipAsserts lead to crashes and this information makes it possible to fix many of those. Collects ShipAsserts from the product, which helps to identify errors.

The following fields are collected:

- **Count** – The count of each assert reported
- **EndTime** – Time at which the last assert reported occurred
- **ErrorGroup** – A bucketing identifier for each assert
- **FirstTimeStamp** – The first time at which the assert occurred
- **Trackback** – A unique identifier for a specific assert

Office.System.SystemHealthErrorsEtwShim

Used to identify customer impacting issues within the running app that may manifest as crashes or degraded functionality. Records errors that occur during process run time.

The following fields are collected:

- **EndTime** – The time at which the last error reported occurred
- **Trackback** – A unique identifier for a specific error
- **ErrorGroup** – A bucketing identifier for each error
- **Count** – The count of each error
- **FirstTimeStamp** – The first time at which the error occurred

Office.System.SystemHealthErrorsUlsAndAsserts

Used to identify customer impacting issues within the running app that may manifest as crashes or degraded functionality. Records errors that occur during process run time.

The following fields are collected:

- **EndTime** – The time at which the last error reported occurred
- **Trackback** – A unique identifier for a specific error
- **ErrorGroup** – A bucketing identifier for each error
- **Count** – The count of each error
- **FirstTimeStamp** – The first time at which the error occurred

Office.System.SystemHealthErrorsUlsWorkaround

Used to identify customer impacting issues within the running app that may manifest as crashes or degraded functionality. Records errors that occur during process runtime

The following fields are collected:

- **EndTime** – The time at which the last error reported occurred
- **Trackback** – A unique identifier for a specific error
- **ErrorGroup** – A bucketing identifier for each error
- **Count** – The count of each error

Office.System.SystemHealthErrorsWithoutTag

Used to identify customer impacting issues within the running app that may manifest as crashes or degraded functionality. Records errors that occur during process runtime.

The following fields are collected:

Count - The count of each error

- **EndTime** - The time at which the last error reported occurred
- **ErrorCode** – An identifier for the error
- **ErrorGroup** - A bucketing identifier for each error
- **ErrorId** – An identifier for the error
- **FirstTimeStamp** - The first time at which the error occurred
- **Trackback** - A unique identifier for a specific error

Office.System.SystemHealthErrorsWithTag

Used to identify customer impacting issues within the running app that may manifest as crashes or degraded functionality. Records errors that occur during process runtime.

The following fields are collected:

- **Count** - The count of each error
- **EndTime** - The time at which the last error reported occurred
- **ErrorCode** – An identifier for the error
- **ErrorGroup** - A bucketing identifier for each error
- **ErrorId** – An identifier for the error
- **FirstTimeStamp** - The first time at which the error occurred
- **Trackback** - A unique identifier for a specific error

RenewIdentityFailure

Collected when a user attempts to open an IRM protected doc or apply IRM protections. It contains the information needed to be able to properly investigate and diagnose issues that happen when failed to renew user certificates.

The following fields are collected:

- **AppInfo.ClientHierarchy** - Client hierarchy, which indicates the application runs in production environment or developer environment
- **AppInfo.Name** - Application name.
- **AppInfo.Version** - Application version
- **Failure.Category** - The category of the failure "UnhandledError"
- **Failure.Detail** - The detailed info of the failure
- **Failure.Id** - Failure ID
- **Failure.Signature** - The signature of the failure, which is same as the event name
- **iKey** - Logging service server ID
- **RMS.HRESULT** - The result of renewing user certificate
- **RMS.ScenarioId** - Scenario ID defined by Rights Management Service Client
- **RMS.SDKVersion** - The version of Rights Management Service Client

save.error

Allows us to detect and fix situations where there was an error when you attempted to save a file. It tracks errors caused by failures to save a file, including a descriptive error message to help us resolve the issue.

The following fields are collected:

- **error** - The type of error that happened to help us detect and resolve issues related to a specific error type
- **file_type** - The type of file the user tried to save (such as .doc)
- **origin** - Where the file save attempt originated from (such as from an email) so we can detect issues associated with saving a file from a specific place in the app
- **token_type** - the type of token used to authenticate the account in order to save the file to help us detect authentication issues associated with saving a file

wkwebview.error

This event lets us detect when web view errors occurred when composing or reading email so that we can prevent issues that could cause your app can't compose email or read email.

The following fields are collected:

- **description** - description for the error
- **error_code** - error code for WKError
- **function_name** - JavaScript function name when error
- **js_exception_column_number** - The column number where JavaScript exception occurred
- **js_exception_line_number** - The line number where JavaScript exception occurred
- **js_exception_message** - The exception message when JavaScript exception occurred
- **js_exception_source_url** - The source URL where JavaScript exception occurred
- **scenario** - where the error occurred. It's an enum. Possible values are old_renderer, react_renderer, and composing.

wkwebview.terminate

This event allows us to detect when web view is terminated by the system. This data allows us to monitor the error user encountered when composing or reading an email.

The following fields are collected:

- **is_foreground** - whether app is in foreground when this event happens.
- **Scenario** - where the error occurred, when rendering or composing.

Device connectivity and configuration data events

The following are the data subtypes in this category:

- Device connectivity and configuration

Device connectivity and configuration subtype

Network connection state and device settings, such as memory.

application.did.receive.memory.warning

This event is sent when Apple tells us that the application is running out of memory. It tells us that we have introduced an issue with memory management on your device.

The following fields are collected:

- **current_memory_used** - Tells us the amount of memory used by the application at the point the application has run out of memory.
- **current_memory_used_percentage** - Tell us the percentage of memory used by the application out of the total memory available at the point the application has run out of memory.
- **currentVC** - Tells us the view that is currently showing when the application has run out of memory.
- **has_hx** - Tells us the account is using our new sync service to help us detect issues caused by our sync service
- **is_watch_app_installed** - Tells us if the user is currently using an Apple Watch and whether it is installed to help us understand the negative performance impact due to the Watch
- **is_watch_paired** - Tells us if the user is currently using an Apple Watch and whether it is paired with the device to help us understand the negative performance impact due to the Watch
- **is_watch_supported_and_active** - Tells us if the user is currently using an Apple Watch and whether it is active to help us understand the negative performance impact due to the Watch
- **rn_initialized** - Tell us if React Native has been initialized at the point the application has run out of memory.
- **running_time** - Tell us the amount of time that app has spent running at the time the application has run out of memory.

conversation.memory.leak

Allows us to detect situations where our email conversation view is causing us to use up more memory on your device that expected.

The following fields are collected:

- No fields or added data are collected. Only logs are collected if there is a memory leak related to a conversation thread.

core.data.corruption

Allows us to detect situations where we cannot show you your email or calendar because where we store your email on your device has become corrupted.

The following fields are collected:

- **errorSource** - indicates whether it came from a save or create action
- **sqlError** - numerical error code listed at https://www.sqlite.org/c3ref/c_abort.html

core.data.corruption.user.reset

Allows us to detect situations where you have deleted or reset your account in our app and it was caused by a corruption in email data we've stored on your device.

The following fields are collected:

- **errorSource** - dictates where the corruption occurred whether during save or create

core.data.diagnostics

Allows us to detect and fix situations where our email storage is using up too much of your device storage space

The following fields are collected:

- **db_size_megabytes** - tracks the size of the core data database rounded to the nearest 25 megabytes and with a maximum megabyte of 500

general.properties.log

This event collects information that allows us to categorize and classify issues within the Outlook app that are related to accessibility and device settings. This categorization is necessary to prioritize the impact of issues on customers.

The following fields are collected for iOS only:

- **alternate_app_icon**- Tell us the alternate app icon that user currently selected by the application
- **bold_text** - Tells us if the device has bold text turned on to help us detect issues related to bold text
- **closed_captioning** - Tells us if the user has turned on closed captioning on their device to help us detect issues related to closed captioning
- **darker_system_colors** - Tells us if the user has turned on darkening of system colors on their device to help us detect issues related to this setting
- **gray_scale** - Tells us if the user has turned on gray scale on their device to help us detect issues related to this setting
- **guided_access** - Tells us if the user has turned on guided access on their device to help us detect issues related to this setting
- **invert_colors** - Tells us if the user has turned on the setting to invert colors on their device to help us detect issues related to this setting
- **mono_audio** - Tells us if the user has turned on the setting for mono audio on their device to help us detect issues related to this setting
- **reduce_motion** - Tells us if the user has turned on the setting for reducing motion on their device to help us detect issues related to this setting
- **reduce_transparency** - Tells us if the user has turned on the setting to reduce transparency on their device to help us detect issues related to this setting
- **speak_screen** - Tells us if the user has turned on the setting for mono audio on their device to help us detect issues related to this setting
- **speak_selection** - Tells us if the user has turned on the setting for Speak Selection on their device to help us detect issues related to this setting
- **switch_control** - Tells us if the user has turned on the setting for Switch Control on their device to help us detect issues related to this setting
- **voice_over** - Tells us if the user has turned on the setting for voiceover on their device to help us detect issues related to this setting

The following fields are collected for Android only:

- **braille** - Tells us if the user has turned on the setting to invert colors on their device to help us detect issues related to this setting
- **caption** - Tells us if the user has turned on closed captioning on their device to help us detect issues related to closed captioning
- **color_inversion** - Tells us if the user has turned on the setting to invert colors on their device to help us detect issues related to this setting
- **density_setting** - The custom (user-selected) density mode currently in use by the application
- **high_contrast** - Tells us if the user has turned on the setting for high contrast on their device to help us

detect issues related to this setting

- **large_text** - Tells us if the device has large text setting turned on to help us detect issues related to this setting
- **oem_preinstall** - Tells us if our app was pre-installed on the device (this applies to Samsung devices only)
- **supported_abis** - Tells us what kind of application binary interfaces (ABIs) are supported by the device platform to help us detect issues related to this setting
- **switch_access** - Tells us if the user has turned on the setting for Switch Access on their device to help us detect issues related to this setting
- **talkback** - Tells us if the user has turned on the setting for talkback on their device to help us detect issues related to this setting
- **theme_color** - The custom (user-selected) theme color currently in use by the application
- **webview_kernel_version**: The Chromium kernel version of webview on the device to help us detect compatibility issues related to the version of webview.
- **webview_package_name**: The package name of webview on the device to help us detect compatibility issues related to the version of webview.
- **webview_package_version**: The package version of webview on the device to help us detect compatibility issues related to the version of webview.

low.storage.warning

This is needed to monitor if our app suddenly takes up most of your device storage due to high memory usage by indicating when the device is low on storage

The following fields are collected:

- **free_bytes** - the amount of free storage space available on the device

Office.AirSpace.AirSpaceLocalBlocklistDriverUpdated

User has updated a video card driver that was previously causing Office crashes and thus no longer being used to render. Informs Microsoft that users who were once in a suboptimal rendering state are once again in the recommended rendering state.

The following fields are collected:

- **Data_BlockedDriverVersion** - Version of the driver that was blocklisted.
- **Data_DeviceId** -identifier of the video card device that was blocklisted.
- **Data_UpdatedDriverVersion** - Version of the updated driver

Office.AirSpace.AirSpaceLocalBlocklistInfo

Details on the user's video card driver that has caused multiple recent crashes of Office applications. Office will not use this video card in this Office session (using software rendering instead) until the driver is updated. Informs Microsoft of video card drivers that are causing problems in Office so trends can be identified and the user of impact of such drivers can be analyzed. Tell Microsoft how many users are in this suboptimal state.

The following fields are collected:

- **Data_AllAppsBlocked** - Whether all Office apps are blocklisted
- **Data_BlockedDeviceId** - identifier of the video card device that was blocklisted
- **Data_BlockedDriverVersion** - Version of the driver that was blocklisted

- **Data_CrashHistory** - A string that represents the history of video card driver caused crashes for analysis
- **Data_SecsBetweenCrashes** - How frequently driver card crashes are occurring

Office.AirSpace.AirSpaceWinCompIsEnabled

Whether the latest Office low-level rendering platform based on Windows Composition is being used.

As the latest Office low-level rendering platform is developed and begins to be released to customers, this allows Microsoft to see how many users are on each version to ensure the platform remains bug-free.

The following fields are collected:

- **Data_WinCompEnabled** -Whether the Windows Composition-based backend is in use

Office.AirSpace.Backend.Win32.GraphicsDriverHangDetectorBlocklistApp

User's video card has been detected as causing long or unrecoverable hangs. Office will not use this video card in this Office session (using software rendering instead) until the driver is updated. Informs Microsoft of video card drivers that are causing problems in Office so trends can be identified and the user of impact of such drivers can be analyzed. Also helps in informing how many users are in this suboptimal state.

The following fields are collected:

- **Data_AppName** - Which app has encountered video card driver hangs

Office.AirSpace.Backend.Win32.GraphicsDriverHangDetectorRegistryWrite

Office has identified that the user's video card driver has caused a hang that should be analyzed at the next Office application boot. Used to determine whether using a different video card driver or adapter would offer a better user experience. As patterns occur, Microsoft may make adjustments to keep the Office experience as smooth as possible.

The following fields are collected:

- **Data_HangDetected** - Whether a hang was detected
- **Data_InDeviceCall** - Which video card rendering call Office was in when the hang occurred
- **Data_Timeout** - How long the hang lasted, if it recovered
- **Data_UnrecoverableCommand** - Whether the hang in this video card rendering command is typically recoverable.

Office.AirSpace.Backend.Win32.LocalBlocklistActivity

Details on the user's video card driver that has caused multiple recent crashes of Office applications. Office will not use this video card in this Office session (using software rendering instead) until the driver is updated. Informs Microsoft of video card drivers that are causing problems in Office so trends can be identified and the user of impact of such drivers can be analyzed. Tell Microsoft how many users are in this suboptimal state.

The following fields are collected:

- **Data.AllAppsBlocked** - Whether all Office apps are blocklisted
- **Data.BlockedDeviceId** - identifier of the video card device that was blocked
- **Data.BlockedDriverVersion** - Version of the driver that was blocklisted
- **Data.CrashHistory System.String** - A string that represents the history of video card driver caused crashes for analysis
- **Data.SecsBetweenCrashes** - How frequently driver card crashes are occurring

Office.AirSpace.Backend.Win32.LocalBlocklistDriverUpdatedActivity

User has updated a video card driver that was previously causing Office crashes and thus no longer being used to render. Informs Microsoft that users who were once in a suboptimal rendering state are once again in the recommended rendering state.

The following fields are collected:

- **Data_BlockedDeviceId** - identifier of the video card device that was blocklisted
- **Data_BlockedDriverVersion** - Version of the driver that was blocklisted
- **Data_UpdatedDriverVersion** - Version of the updated driver

Office.Graphics.SpriteMemCorrupt

Reports any errors detected in the sprite memory accounting telemetry. Critical for assessing health of the graphics memory usage telemetry. This information is needed to validate the correctness of our SpriteMem telemetry.

The following fields are collected:

- **Data_CurrentSpriteMem** - Total amount of memory that is actively allocated to hold sprites (images) that result in screen content.
- **Data_Function** - The name of the function that is attempting to release sprite memory.
- **Data_SpriteMemToRemove** - Amount of memory to be removed from sprite allocation.

Office.PowerPoint.PPT.Shared.NoInternetConnectivity

Collected whenever PowerPoint detects there is no internet connectivity. Microsoft uses this data to get diagnostic information about the user's internet connection to be able to understand how that impacts connectivity to Office services.

The following fields are collected:

- **Data_IsNexusDetected:bool** - shows whether we got the Internet connectivity status when calling Nexus service (value true) or when calling generic web service API call (value false)

Office.ServiceabilityManager.OfficeSvcMgrProfile

This event is triggered when Office Serviceability Manager starts, and is critical for providing accurate insights related to Deployment Status and Application and Add-in crashes within customer's tenant by allowing us to generate insights for the IT Admin to be able to confidently roll out updates for their enterprise machines.

The following fields are collected:

- **DeviceIdJoinToken** - used to join Telemetry data from Health and Deployment Status with other Functional Data, which is collected via the Services pipeline.
- **TenantAssociationKeyStamped** - a Boolean flag used to determine the number of Managed devices in the Office eco-system.

Optional diagnostic data for Office

8/25/2021 • 5 minutes to read • Edit Online

> **NOTE**
> For a list of Office products covered by this privacy information, see Privacy controls available for Office products.

Diagnostic data is used to keep Office secure and up-to-date, detect, diagnose and fix problems, and also make product improvements. This data does not include a user's name or email address, the content of the user's files, or information about apps unrelated to Office.

This diagnostic data is collected and sent to Microsoft about Office client software running on the user's device. Some diagnostic data is required, while some diagnostic data is optional. We give you the ability to choose whether to send us required or optional diagnostic data through the use of privacy controls, such as policy settings for organizations. You can see the diagnostic data being sent to us by using the Diagnostic Data Viewer.

> **NOTE**
> If you're using a version of Office 2019 or Office 2016 that doesn't give you or your admin the ability to choose whether to send us required or optional diagnostic data, then only required diagnostic data is sent. For example, if you're using Office Professional Plus 2019 or Office Standard 2016, which don't provide that choice, then only required diagnostic data is sent. Office 2013 doesn't send required or optional diagnostic data. For more information about which versions of Office provide this choice, see Privacy controls available for Office products.

Optional diagnostic data is additional data that helps us make product improvements and provides enhanced information to help us detect, diagnose, and fix issues.

If you choose to send us optional diagnostic data, required diagnostic data is also included. Also, diagnostic log files for Office, which contain information very similar to optional diagnostic data, might be sent. For more information about those log files, see Overview of diagnostic log files for Office.

Examples of optional diagnostic data include data we collect about the shapes users insert into Word documents so we can provide better options, and data we collect about the time it takes for a PowerPoint slide to appear on your screen so we can improve the experience if it's slow.

For more information about diagnostic data, see the following articles:

- Required diagnostic data for Office
- Using the Diagnostic Data Viewer with Office

If you're the admin for your organization, you might also be interested in the following articles:

- Overview of privacy controls for Microsoft 365 Apps for enterprise
- Use policy settings to manage privacy controls for Microsoft 365 Apps for enterprise
- Use preferences to manage privacy controls for Office for Mac
- Use preferences to manage privacy controls for Office on iOS devices
- Use policy settings to manage privacy controls for Office on Android devices

Categories of optional diagnostic data

Optional diagnostic data is organized into the following categories:

- Software setup and inventory
- Product and service usage
- Product and service performance
- Device connectivity and configuration

These categories are shown in the Diagnostic Data Viewer and are the same categories used with required diagnostic data.

The following sections provide a description of each category and examples of events for each category.

Software setup and inventory events

This category includes events that may cover the following areas:

- Installed product and version and the installation status
- Software add-ins and their settings.
- Document, feature, and add-in error conditions that may compromise security, including product update readiness.

The following table provides examples of events in this category and a description of those events.

EVENT NAME	EVENT DESCRIPTION
Office.Extensibility.AppCommands.GetRibbonUpdatesForUserId	This event indicates whether Word successfully updates the Ribbon in the Word User Interface when the user changes their identity. We use this event to detect incorrect setup and other issues that would affect the Office user interface.
Office.Extensibility.AppCommands.AppCmdInstall	This event provides information about the Office add-in that the user has installed, including app ID, operating system build and version, success of installation, and duration of install.

Product and service usage events

This category includes events that may cover the following areas:

- Success of application functionality. Limited to opening and closing of the application and documents, file editing, and file sharing (collaboration).
- Determination if specific feature events have occurred, such as start or stop, and if feature is running.
- Office accessibility features

The following table provides examples of events in this category and a description of those events.

EVENT NAME	EVENT DESCRIPTION
Office.Word.Commanding.Highlight	This event indicates Word has executed the command to highlight text. We use this event to detect errors in the text-highlight command.
Office.Translator.AddInLoaded	A heartbeat to indicate that the translator feature has been loaded and rendered successfully.

EVENT NAME	EVENT DESCRIPTION
Office.Graphics.GVizInsertShape	Tracks the usage of the Insert Shape feature in Word and also reports details of types of shapes inserted and from which source.
Office.PowerPoint.PPT.Desktop.SummaryZoomInsertionRule	This event determines if there are any sections present in a document when the user is inserting Summary Zoom and if the user chooses to delete existing sections.
Office.Security.SecureReaderHost.ProtectedViewValidation	Tracks when and why a file is opened in Protected View. Used to diagnose conditions where Protected View may not be correctly triggered to ensure the feature is working properly.

Product and service performance events

This category includes events that may cover the following areas:

- Unexpected application exits (crashes) and the state of the application when that happens.
- Poor response time or performance for scenarios such as application start up or opening a file.
- Errors in functionality of a feature or user experience.

The following table provides examples of events in this category and a description of those events.

EVENT NAME	EVENT DESCRIPTION
Office.Word.Word.CoreSaveTime100ns	This event logs the performance of a document save activity by Word. We use this event to detect errors and performance issues in the Word save document activity.
Office.Identity.SignInForWamAccountAad	This event is sent when a user is signed in to an Azure Active Directory account with Web Account Manager (WAM) library. This event sends metadata such as AppName, AppVersion, and ErrorCode if the event failed.
Office.PowerPoint.PPT.Desktop.FileOpen.FirstSlideMasterThumbnailRenderTime	This event collects the length of time it takes to render the first slide master thumbnail in PowerPoint.
Office.Extensibility.Diagnostics	This event provides general diagnostic information for Office add-ins, such as crash reports for debugging.

Device connectivity and configuration events

This category includes events that may cover the following areas:

- Network connection state and device settings, such as memory.

The following table provides examples of events in this category and a description of those events.

EVENT NAME	EVENT DESCRIPTION
Office.Graphics.ArtViewValidate	This event logs validation the results of Graphics View that supports Graphics User Interface. We use the event to collect usage and error data about graphics rendering.

EVENT NAME	EVENT DESCRIPTION
Office.Graphics.ARCExceptionScope	This event tracks rendering failures coming from the rendering engine.
Office.Extensibility.ODPLatency	This event provides information about the user's network connection and speed.

Connected experiences in Office

9/1/2021 • 10 minutes to read • Edit Online

NOTE

For a list of Office products covered by this privacy information, see Privacy controls available for Office products.

Office consists of client software applications and connected experiences designed to enable you to create, communicate, and collaborate more effectively. Working with others on a document stored on OneDrive or translating the contents of a Word document into a different language are examples of connected experiences.

Connected experiences that analyze your content

Connected experiences that analyze your content are experiences that use your Office content to provide you with design recommendations, editing suggestions, data insights, and similar features. For example, PowerPoint Designer or Translator.

The following table provides a list of connected experiences that analyze your content and also provides links to more information about them.

NAME	MORE INFORMATION
*3D Maps**	Get started with 3D Maps
Analyze Data (previously named Ideas)	Analyze Data in Excel
Automatic alt txt	Everything you need to know to write effective alt text
Chart recommendations	Create a chart with recommended charts
Class Notebook add-ins (OneNote)	Getting Started with the OneNote Class Notebook: A Walkthrough for Teachers
Data types	Excel data types: Stocks and geography
Dictation	Dictate in Microsoft 365
Editor[1]	Microsoft Editor checks grammar and more in documents, mail, and the web
Friendly links in Outlook	Friendlier link names in email
Ink to Text, Ink to Shape, Ink to Math[2]	Change handwritten ink to shapes, text, or math in PowerPoint for Microsoft 365
Insert data from picture	Insert data from picture
Learning Tools	Learning Tools

NAME	MORE INFORMATION
Live captions & subtitles	Present with real-time, automatic captions or subtitles in PowerPoint
*Map chart**	Create a Map chart in Excel
Maps in Power View	Maps in Power View
Office Presentation Service	Broadcast your PowerPoint presentation online to a remote audience
PivotTable recommendations	Create a PivotTable to analyze worksheet data
PowerPoint Designer	Create professional slide layouts with PowerPoint Designer
Presenter Coach (PowerPoint)	Rehearse your slide show with Presenter Coach
Publish to Microsoft Stream	Turn your presentation into a video
Publish to Power BI	Publish to Power BI from Excel
*Research**	Add or change research services
Scan a business card	Scan or tap to add contacts on Outlook for Android
Sharing notification	Get notified when members of your team update your shared file
*Similarity checker** (a feature of Editor)	Check your document for similarity to online sources
*Smart Lookup**	Get insights into what you're working on with Smart Lookup
Suggested replies	Use suggested replies in Outlook
Tap for Word (Add from files)	Find and use the content you need, when you need, without leaving Word
Text predictions (a feature of Editor)	Make writing faster with text predictions in Word
Transform to Web Page	Transform your Word document into a Sway web page
Translator	Translate text into a different language

[1] Basic functionality of Editor remains available even if you're not connected to the internet. That same basic functionality also remains available if you decide to disable the use of connected experiences that analyze your content. In both cases, no data about the use of Editor is sent to Microsoft.

[2] For devices running Android, these ink capabilities remain available even if you're not connected to the internet. These ink capabilities also remain available on devices running Android even if you decide to disable the use of connected experiences that analyze your content. In both cases, no data about the use of these ink capabilities is sent to Microsoft.

NOTE

*When users are signed in with a work or school account, the connected experiences listed above in italics are optional and are provided under the terms of the Microsoft Services Agreement and privacy statement, and other terms may also apply. For more information, see Overview of optional connected experiences in Office. If you're an admin, these connected experiences can be managed by privacy controls for connected experiences or by the *Allow the use of additional optional connected experiences in Office* policy setting.

Connected experiences that download online content

Connected experiences that download online content are experiences that allow you to search and download online content including templates, images, 3D models, videos, and reference materials to enhance your documents. For example, Office templates or PowerPoint QuickStarter.

The following table provides a list of connected experiences that download online content and also provides links to more information about them.

NAME	MORE INFORMATION
Calendar apps	What are Calendar Apps?
Cloud fonts	Cloud fonts in Office
FAQs	When you choose **Settings** > **Help & Feedback** > **FAQs** on Outlook for iOS and Android, you are taken to either Outlook for iOS Help or Outlook for Android Help.
Ink Effects	Draw and write with ink in Office
Insert Icons	Insert icons in Microsoft Office
Insert Microsoft Forms	Insert a form or quiz into PowerPoint
*Insert Online 3D Models**	Get creative with 3D models
*Insert Online Pictures**	Insert pictures
*Insert Online Video**	Insert a video from YouTube or another site
*Interesting Calendars**	Subscribe to a calendar about your favorite sports teams or TV shows on Outlook for iOS and Android.
*Location suggestions**	When you add a public place with a street address to a calendar item, Outlook for iOS and Android will also include the full street address in the Location field.
Maps in event details	The event details page will show a map based on the address of the location of the event on Outlook for iOS and Android.
Office Help	When you choose **Help** > **Help** on the ribbon or use F1 in an Office app
Online Shape Search (Visio)	Find more shapes and stencils

NAME	MORE INFORMATION
*PowerPoint QuickStarter**	Research a topic with PowerPoint QuickStarter
*Researcher**	Research your paper easily within Word
Tell Me[3]	Do things quickly with Tell Me
Templates	Download free, pre-built templates
*Travel time estimates in Up Next**	When you choose to allow Outlook for iOS and Android to access your location, an estimate of your travel time to your next event will be included in the Up Next card at the top of your inbox. What are Time to Leave notifications?
*Weather Bar** (Outlook)	Change the calendar Weather Bar forecast city
*Weather on calendar** (Outlook mobile)	How do I turn on weather on calendar on Outlook mobile?

[3] Basic functionality of Tell Me remains available even if you're not connected to the internet. That same basic functionality also remains available if you decide to disable the use of connected experiences that download online content. In both cases, no data about the use of Tell Me is sent to Microsoft.

> **NOTE**
> *When users are signed in with a work or school account, the connected experiences listed above in italics are optional and are provided under the terms of the Microsoft Services Agreement and privacy statement, and other terms may also apply. For more information, see Overview of optional connected experiences in Office. If you're an admin, these connected experiences can be managed by privacy controls for connected experiences or by the *Allow the use of additional optional connected experiences in Office* policy setting.

Other connected experiences

In addition to connected experiences that analyze content and connected experiences that download online connected experiences, there are some other connected experiences available in Office.

The following table provides a list of those other connected experiences and provides links to more information about them.

NAME	MORE INFORMATION
@mention	Use @mention in comments to tag someone for feedback
Brainstorming diagram (Visio)	Create a brainstorming diagram
Coming Soon (Outlook)	Coming Soon
Contact Support (Outlook)	When you choose **Help** > **Contact Support** on the ribbon
Custom Import (Visio)	Import data to shapes in your drawing

NAME	MORE INFORMATION
Data Loss Prevention (DLP) policy tips	Send email notifications and show policy tips for DLP policies
Data Visualizer (Visio)	Create a Data Visualizer diagram
Design Flows (Visio)	Design an automated workflow in Visio
External list (Outlook)	Connect an external list to Outlook
Focused Inbox (Outlook)	Focused Inbox for Outlook
Information Rights Management	Open a file that has restricted permissions
LinkedIn contact card	LinkedIn in Microsoft apps and services
*Office add-ins**	View, manage, and install add-ins in Office programs *(for users)* Deploy add-ins in the admin center *(for admins)*
Office feedback	How do I give feedback on Microsoft Office?
Office support (Contact us)	When you choose **Help** > **Contact Support** on the ribbon
Organization chart (Visio)	Create an organization chart in Visio
*Outlook UserVoice**	How do I give feedback on Microsoft Office?
PivotDiagram (Visio)	Create a PivotDiagram in Visio
Recent documents	Open files from the File menu
*Resume Assistant (or CV Assistant)**	Use Resume Assistant and LinkedIn for great resumes
Room Finder (Outlook)	How to control the Room Finder in Outlook
Safe Documents	Safe Documents Safe Documents in Microsoft 365 E5
Safe Links	ATP Safe Links
Sensitivity labels	Apply sensitivity labels to your files and email in Office
Share	Share your Excel workbook with others Share a presentation (PowerPoint) Share a document (Word)
Shared with me	See files others have shared with you
SharePoint site mailbox (Outlook)	Show or hide a site mailbox in Outlook
Timeline import and export (Visio)	Import and export timeline data between Visio and Project

NAME	MORE INFORMATION
Version history	View previous versions of Office files
While you were away	Get notified when members of your team update your shared file

NOTE

*When users are signed in with a work or school account, the connected experiences listed above in italics are optional and are provided under the terms of the Microsoft Services Agreement and privacy statement, and other terms may also apply. For more information, see Overview of optional connected experiences in Office. If you're an admin, these connected experiences can be managed by privacy controls for connected experiences or by the *Allow the use of additional optional connected experiences in Office* policy setting.

Choose whether these connected experiences are available to use

You can choose whether certain types of connected experiences, such as connected experiences that download online content, are available to use. How you make that choice depends on whether you are signed into Office with a Microsoft account, such as a personal outlook.com email address, or with a work or school account.

If you are signed in with a Microsoft account, open an Office app, such as Word, and go to **File** > **Account** > **Account Privacy** > **Manage Settings**. Under the **Connected experiences** section, you can choose whether certain types of connected experiences, such as experiences that analyze your content, are available to use. If you don't go to **Manage Settings**, all connected experiences will be available to you.

If you are signed in with a work or school account, the admin in your organization will decide whether these connected experiences are available to you. You won't see any choices for these connected experiences if you go to **File** > **Account** > **Account Privacy** > **Manage Settings**.

NOTE

- If you're using Office for Mac, open any Office application and select the app menu (such as Word, or Excel) > **Preferences** > **Privacy**. This will open the Account Privacy settings dialog box where you can see your privacy options.
- For more information, see Account Privacy Settings.

If you are the admin for your organization, you can use policy settings to determine whether these connected experiences are available to your users. If you don't use these policy settings, all connected experiences will be available for your users. For more information about using these policy settings, see the following articles:

- Use policy settings to manage privacy controls for Microsoft 365 Apps for enterprise
- Use preferences to manage privacy controls for Office for Mac.
- Use preferences to manage privacy controls for Office on iOS devices
- Use policy settings to manage privacy controls for Office on Android devices

If you choose to turn off some types of connected experiences, either the ribbon or menu command for those connected experiences will be grayed out or you will get an error message when you try to use those connected experiences.

Even if you decide to turn off these connected experiences, certain Office functionality will remain available, such as synching a mailbox in Outlook, and Teams and Skype for Business will continue to work. Also, there is a set of services that are essential to how Office functions that can't be turned off, such as the licensing service that

confirms that you are properly licensed to use Office.

Related articles

- Overview of privacy controls for Microsoft 365 Apps for enterprise
- Overview of optional connected experiences in Office
- Required service data for Office
- Essential services for Office

Overview of optional connected experiences in Office

8/25/2021 • 9 minutes to read • Edit Online

> **NOTE**
> For a list of Office products covered by this privacy information, see Privacy controls available for Office products.

If you have a work or school account, your organization's admin may have provided you with the ability to use one or more cloud-backed services (also referred to as "optional connected experiences") while using the Office apps, like Word or Excel, that are included with Microsoft 365 Apps for enterprise (previously named Office 365 ProPlus). These cloud-backed services are optional. Whether you use them is up to you. They are provided to you under the terms of the Microsoft Services Agreement and privacy statement. In some cases, other terms may also apply. This article lists the cloud-backed services, further explains their terms of use, and describes how you can turn them off or on at any time.

> **NOTE**
> - If you're an admin, see Admin controls for optional connected experiences.
> - For Microsoft Teams, see Overview of optional connected experiences in Microsoft Teams.

If you use these services, what terms of use do you need to agree to?

It is important to know that these optional cloud-backed services are not covered by your organization's license with Microsoft. Instead, they are licensed directly to you. By using these optional cloud-backed services, you also agree to the terms of the Microsoft Services Agreement and privacy statement. Depending on which service you use, additional terms may also apply. In most cases, there is no fee to use these services. If a fee does apply (as may be the case with some add-ins available to you for download through Office Store), you will be clearly informed prior to use.

Experiences that rely on Bing

Some cloud-backed services are powered by Microsoft Bing. Microsoft Bing is a separate business to Microsoft 365 Apps for enterprise and is managed independently of Microsoft 365 Apps for enterprise by Microsoft. The experiences it powers in Office apps included with Microsoft 365 Apps for enterprise are: 3D Maps, Map Charts, Insert Online Pictures, Insert Online 3D Models, PowerPoint QuickStarter, Researcher, Smart Lookup, and similarity checker (a feature of Editor).

Bing powers the following experiences in Outlook for iOS and Android:

- **Location suggestions:** When you add a public place with a street address to a calendar item, the full street address is included in the Location field.
- **Interesting Calendars:** You can subscribe to calendars about your favorite sports teams or TV shows.
- **Travel time in Up Next:** Your next upcoming event will be summarized at the top of your email inbox in a card called Up Next. In Outlook for Android, this experience requires Bing Maps to provide estimated travel time.
- **Weather on calendar:** The weather forecast for your current location will appear at the top of your calendar screen.

These experiences that rely on Bing are licensed to you under the terms of the Microsoft Services Agreement and covered by the privacy statement. Any search queries you provide to the Microsoft 365 Apps for enterprise organization while using these services are sent to Microsoft Bing. They are not linked to you by the Bing organization.

Experiences that rely on LinkedIn

Resume Assistant provides an experience in Word that is powered by LinkedIn. It delivers ideas for your resume. LinkedIn is a different company that is owned by Microsoft. If you use Resume Assistant, the LinkedIn user agreement and privacy policy apply in addition to the Microsoft Services Agreement and privacy statement. Any search queries you provide to the Microsoft 365 Apps for enterprise organization while using these services are sent to LinkedIn. They are not linked to you by the LinkedIn organization. You do not need to have a LinkedIn account to use this cloud-backed service.

Note: Your privacy settings do not control experiences that require you to connect your LinkedIn account to your Microsoft work or school account. To control these types of experiences (such as the LinkedIn information on a profile card in Outlook), see LinkedIn in Microsoft apps and services.

Experiences that rely on other Microsoft-owned online services and/or services owned by third-parties

Help > Contact Support

On Outlook for iOS and Android, you can report issues and connect with our support team through **Settings** > **Help & Feedback** > **Contact Support**. This experience requires a Microsoft owned service called PowerLift, and the terms of the Microsoft Services Agreement and Microsoft privacy statement apply.

Help > Suggest a Feature

Suggest a Feature is an experience that allows you to submit your ideas regarding your use of Outlook or Excel. Your suggestions are provided directly to a third-party service called UserVoice. If you submit ideas using Suggest a Feature, the terms of service and privacy policy of UserVoice will apply.

Insert online video

Insert Online Video allows you to insert video files into your document. The insert online video experience is subject to the Microsoft Services Agreement and privacy statement. Additional terms of use may apply if you access content from third-party sources. For example, when you connect to YouTube, its terms of use and privacy statement will apply. Microsoft may add other providers of video material in the future.

Microsoft Error Reporting Program (MERP)

MERP connects to the Watson.microsoft.com service to send diagnostic data when Office client applications that run on Mac devices crash. It is owned by Microsoft and the terms of the Microsoft Services Agreement and Microsoft privacy statement apply. MERP does not apply to Office client applications that run on Windows.

Office Store

When you use the Office Store, your use of the store site itself is licensed by the Microsoft Services Agreement and privacy statement. However, any add-ins that you download through the Office Store are covered by the add-in provider's terms of use. These providers can be various different organizations or businesses, and some may charge a usage fee. You will need to check the permissions, privacy statement, and license terms of each add-in to know which terms apply and decide if you want to agree to the terms of use that organization offers.

NOTE

- The administrator in your organization might provide you with add-ins, even if you don't choose to use the Office Store.
- If you're an administrator, for more information about add-ins, see the "Optional connected experiences" section in Privacy and security for Office Add-ins and Privacy, permissions, and security for Outlook add-ins.

Research

Research connects to cloud-backed services to obtain dictionary, thesaurus, translation, and word lookup results. When you use this experience, it connects to Microsoft-owned services by default and the Microsoft Services Agreement and privacy statement apply. Research allows you to add other service providers. If you decide to use another service provider, additional terms may apply.

NOTE

Research is different than Researcher. In newer versions of Office, another experience called Researcher is available in addition to Research. Researcher is a newer service, powered by Bing, and is discussed in the Experiences that rely on Bing section.

Travel time in Up Next in Outlook for iOS

On Outlook for iOS, your next upcoming event will be summarized at the top of your email inbox in a card called Up Next. This experience requires an Apple owned service called Apple Maps to provide estimated travel time. The Apple Maps terms of use and Apple privacy policy apply.

Weather Bar in Outlook

The Weather Bar in Outlook displays weather forecasts for geographic locations you choose. During use, the Weather Bar in Outlook connects to MSN Weather by default. MSN is owned by Microsoft and the terms of the Microsoft Services Agreement and Microsoft privacy statement apply. When you use the Weather Bar with MSN Weather, Microsoft does not detect your location and the cities that you choose to display are not linkable to you. You may change this default setting to connect to other weather service providers. If you decide to use another weather service provider, you will need to check the privacy statement and license terms of that provider to know which terms apply.

How to determine if your admin has given you the ability to use optional connected experiences

To determine whether your admin has given you the ability to use any optional connected experiences in your Office apps included with Microsoft 365 Apps for enterprise on a Windows device, go to **File** > **Account** > **Account Privacy** and select **Manage Settings**. If your admin has not given you control, you will see a message that states, "Your organization's admin manages your privacy settings and has decided to disable optional connected experiences."

If you're using Office for Mac, open any Office application and select the app menu (such as Word, or Excel) > **Preferences** > **Privacy**. This action will open the Account Privacy settings dialog box where you can see your privacy options.

For more information, see Account Privacy Settings.

> **NOTE**
> If the admin for your organization has provided you with a volume licensed version of Office 2019, Project 2019, or Visio 2019, go to **File** > **Options** > **Trust Center** > **Trust Center Settings** > **Privacy Options**. There you should see a check box for **Turn on optional connected experiences**. If you don't see that check box in Project 2019 or Visio 2019, choose **Privacy Settings...** in that **Trust Center** dialog box. Then you should see a **Privacy Settings** dialog box appear with a check box for **Enable optional connected experiences**.

Your privacy settings

If your admin has given you the ability to use optional connected experiences in your Office apps included with Microsoft 365 Apps for enterprise on a Windows device, you can go to **File** > **Account** > **Account Privacy** and select **Manage Settings** to manage your settings at any time. When enabled, you will have the option to use the services described in this article when you want to use them. When disabled, they will not be accessible.

If you're using Office for Mac, open any Office application and select the app menu (such as Word, or Excel) > **Preferences** > **Privacy**. This action will open the Account Privacy settings dialog box where you can see your privacy options.

For more information, see Account Privacy Settings.

> **NOTE**
> If the admin for your organization has provided you with a volume licensed version of Office 2019, Project 2019, or Visio 2019, go to **File** > **Options** > **Trust Center** > **Trust Center Settings** > **Privacy Options**. There you should see a check box for **Turn on optional connected experiences**. If you don't see that check box in Project 2019 or Visio 2019, choose **Privacy Settings...** in that **Trust Center** dialog box. Then you should see a **Privacy Settings** dialog box appear with a check box for **Enable optional connected experiences**.

Required service data

When you use any of the optional cloud-backed services described in this article, Microsoft may collect required service data (such as usage data, error and performance data) about the performance of the experience when you used it. This required service data may contain "personal data" as defined by Article 4 of the European GDPR. All required service data Microsoft collects during the use of any Microsoft 365 Apps for enterprise applications and services is pseudonymized as defined in ISO/IEC 19944:2017, (section 8.3.3) standard.

Admin controls for optional connected experiences

If you're an admin, see the following articles to learn how to give or restrict your users' ability to use optional connected experiences:

- Use policy settings to manage privacy controls for Microsoft 365 Apps for enterprise
- Use preferences to manage privacy controls for Office for Mac
- Use preferences to manage privacy controls for Office on iOS devices
- Use policy settings to manage privacy controls for Office on Android devices
- Use policy settings to manage privacy controls for Office for the web applications

Most optional connected experiences also can be managed by the privacy controls for connected experiences. For example, Insert Online Pictures can also be managed by the *Allow the use of connected experiences in Office that download online content* policy setting.

For more information, see Overview of privacy controls for Microsoft 365 Apps for enterprise.

Required service data for Office

8/25/2021 • 3 minutes to read • Edit Online

> **NOTE**
> For a list of Office products covered by this privacy information, see Privacy controls available for Office products.

Office consists of client software applications and connected experiences designed to enable you to create, communicate, and collaborate more effectively. Working with others on a document stored on OneDrive for Business or translating the contents of a Word document into a different language are examples of connected experiences.

As you use a connected experience, data is sent to and processed by Microsoft to provide you that connected experience. This data is crucial because this information enables us to deliver these cloud-based connected experiences. We refer to this data as required service data.

Required service data can include information related to the operation of the connected experience that is needed to keep the underlying service secure, up to date, and performing as expected. If you choose to use a connected experience that analyzes your content, for example Translate in Word, the text you typed and selected to translate in the document is also sent and processed to provide you the connected experience. Required service data can also include information needed by a connected experience to perform its task, such as configuration information about the Office app.

Example of required service data for a connected experience

Let's use PowerPoint Designer as another example to show the type of required service data that a connected experience sends to Microsoft. PowerPoint Designer helps you improve your slides by automatically generating design ideas to choose from. While you're putting content on a slide, Designer works in the background to match that content to professionally designed layouts.

The required service data that PowerPoint Designer sends to Microsoft could include the following information:

- The text or images you added to your slide.
- Which slide you're working on and the slide's layout.
- Whether the design idea was correctly applied to your slide.
- Whether the interaction between PowerPoint and the Designer service performed as expected.

This required service data helps ensure that PowerPoint Designer is performing as expected.

Manage required service data

We give you the ability to choose which types of connected experiences you want to use in Office, which then determines what required service data is sent to Microsoft. Dictation in Word, for example, is one of several connected experiences that analyzes your content. If you choose to turn off connected experiences that analyze content, no required service data about Dictation in Word is sent to Microsoft because Dictation in Word won't be available to use. For more information, see Choose whether these connected experiences are available to use.

Required service data is separate from required or optional diagnostic data, which relates to information about the use of Office software running on your device. Therefore, the privacy settings you chose for required or optional diagnostic data don't affect whether required service data is sent to Microsoft.

Required service data is also collected and sent to Microsoft for the essential services of Office, such as the licensing service that confirms that you're properly licensed to use Office. While you can control many of the connected experiences that are available to you, or to your users if you're the administrator in your organization, this set of services are essential to how Office functions, and therefore cannot be disabled. The data for essential services is always sent to and processed by Microsoft when using Office, regardless of how other privacy-related settings are configured.

Required service data is available through Data Service Requests (DSRs). For more information, see the Microsoft Privacy Statement and Office 365 Data Subject Requests for the GDPR and CCPA.

If you're the administrator for your organization and want to manage connected experiences, see the following articles:

- Overview of privacy controls for Microsoft 365 Apps for enterprise
- Use policy settings to manage privacy controls for Microsoft 365 Apps for enterprise
- Use preferences to manage privacy controls for Office for Mac
- Use preferences to manage privacy controls for Office on iOS devices
- Use policy settings to manage privacy controls for Office on Android devices
- Use policy settings to manage privacy controls for Office for the web applications

Categories of required service data

Required service data is organized into the following categories:

- Software setup and inventory
- Product and service usage
- Product and service performance
- Device connectivity and configuration

The information in these categories enables Microsoft to assess whether a connected experience or essential service is secure, up to date, and performing as expected.

For example, information in the product and service performance category may cover unexpected issues (crashes), poor response times or performance, or errors in functionality.

For the product and service usage category, information might be collected that indicates whether the service used by a connected experience started successfully and was available when needed.

Essential services for Office

8/25/2021 • 448 minutes to read • Edit Online

> **NOTE**
> For a list of Office products covered by this privacy information, see Privacy controls available for Office products.

Office consists of client software applications and connected experiences designed to enable you to create, communicate, and collaborate more effectively. While you can control many of the connected experiences that are available to you, or to your users if you're the admin in your organization, there are a set of services that are essential to how Office functions and therefore cannot be disabled. For example, the licensing service that confirms that you are properly licensed to use Office. Required service data about these services is collected and sent to Microsoft, regardless of any other privacy-related policy settings that you have configured.

For more information, see the following articles:

- Required service data for Office
- Connected experiences in Office

If you're the admin for your organization, you might also be interested in the following articles:

- Overview of privacy controls for Microsoft 365 Apps for enterprise
- Use policy settings to manage privacy controls for Microsoft 365 Apps for enterprise
- Use preferences to manage privacy controls for Office for Mac
- Use preferences to manage privacy controls for Office on iOS devices
- Use policy settings to manage privacy controls for Office on Android devices
- Use policy settings to manage privacy controls for Office for the web applications

List of essential services for Office

The following table provides a list of the essential services for Office and a description of each service.

SERVICE	DESCRIPTION
Authentication	Authentication is a cross-platform service that validates your Office user identity. It is required to enable you to sign in to Office, activate your Office license, access your files stored in the cloud, and provides a consistent experience across Office sessions and your devices.
Click-to-Run	Click-to-Run is the installation technology used to install and update Office on Windows. It checks for new versions of Office, and when a new version is available, downloads and installs it. Click-to-Run will detect the need for, perform the download of, and install Office updates, including security updates.

SERVICE	DESCRIPTION
Enhanced Configuration Service (ECS)	ECS provides Microsoft the ability to reconfigure Office installations without the need for you to redeploy Office. It is used to control the gradual rollout of features or updates, while the impact of the rollout is monitored from diagnostic data being collected. It is also used to mitigate security or performance issues with a feature or update. In addition, ECS supports configuration changes related to diagnostic data to help ensure that the appropriate events are being collected.
Licensing	Licensing is a cloud-based service that supports your Office activation for new installations and maintains the license on your devices after Office has been activated. It registers each of your devices and activates Office, checks the status of your Office subscription, and manages your product keys.
Microsoft AutoUpdate (MAU)	Microsoft AutoUpdate (MAU) is the technology used to update Microsoft applications produced for macOS, such as Office. MAU will detect the need for, perform the download of, and install application updates, including security updates.
OneNote sync	OneNote for Mac only supports notebooks stored on the internet in OneDrive or SharePoint Online. OneNote for Mac continually syncs all of the user's notes with OneDrive or SharePoint Online. This lets users open, view, and edit their notebooks on all their devices so their notebooks are always up to date.
Services Configuration	Services Configuration provides the ability to make updates to Office configuration settings to enable or disable client features. It is called every time an Office application is started and provides details about other Office configurations and services. Services Configuration also controls which services are designated as essential services.
Telemetry	The Telemetry service is used to collect diagnostic data from Office applications. It enables the collection of the diagnostic data generated by Office, both required and optional diagnostic data. It is also responsible for the collection of some required service data for Office.

Events and data fields for essential services for Office

The next sections provide the following information:

- A list of events for each essential service
- A description of each event
- A list of data fields in each event
- A description of each data field

Authentication events

These diagnostic data events are collected when Office attempts to acquire an authentication token, either silently or through prompting.

Office.Android.MSAGuestToAAD

This event helps in understanding how many users are getting prompted for providing personal account

password, while accessing work resource, as their personal account could be a valid guest to work account's tenant.

This data helps us understand how many users are going through the pain of sign in re-prompts to prioritize AAD token acquisition silently based on a Microsoft account SAML (Security Assertion Markup Language) assertion.

The following fields are collected:

- **Tag** - Indicates that user got sign-in prompt for personal account, while accessing work account resource.

Office.Identity.FbaPromptWin32

Collected when Office shows the user a Forms-Based-Auth sign-in prompt.

Along with silent token acquisitions, authentication prompts are used to determine if user is in a broken authentication state which, for the user, results in what is essentially an Offline Client state, or in the worst case, broken authentication may prevent license acquisition and result in a completely unusable client.

Forms-Based-Auth (FBA) sign-in prompts are used for some on-premises authentication scenarios and typically we want to make sure this isn't happening, as everyone should be using Modern-Auth because of security vulnerabilities associated with FBA.

The following fields are collected:

- **AuthScheme** - the authentication scheme used
- **DocumentUrlHash** - an encrypted URL requesting
- **EndTag** - the tag where FBA form is completed
- **Flags** - Obsolete
- **FlowTag** - the tag where FBA form is started
- **LastError** – the error code returned
- **PromptEndTime** - the time when then prompt ended
- **PromptStartTime** - the time when the prompt started
- **Result** - whether the authentication succeeded
- **SessionEndTime** – the time when the event session ended
- **Timeout** - the time when the prompt timed out

Office.Identity.SignOutEvent

Collected when a user signs out of Office.

Knowing the user is signed out makes it possible to classify other events, such as prompts, as expected, so those events can be correctly computed in reliability / ship-readiness metrics and avoid alerting or rolling back builds on the faulty premise that the user is experiencing unexpected sign-in prompts.

The following fields are collected:

- **FlowEndTime** - the time when the sign out action ended
- **FlowStartTime** - the time when the sign out action started
- **IdentityErrorState** - any identity error state during sign out
- **IdentityHashedUniqueId** - the encrypted identity ID being signed out

- **IdentityProviderType** - the identity provider of the identity being signed out
- **IdentityUniqueID** - the identity ID being signed out
- **SessionEndTime** - the time when the event session ended
- **SignOutUserAction** - Indicates the user initiates the sign out action

Office.Identity.SspiPromptWin32

Collected when Office shows the user a Windows SSPI sign-in prompt. Along with silent token acquisitions, authentication prompts determine whether a user is in a broken authentication state, which results in an Offline Client state. Broken authentication may prevent license acquisition and result in a completely unusable client.

Windows SSPI prompts are used for authenticating with Exchange (for mail synchronizing) when the user's Exchange resource hasn't been set up for multi-factor authentication.

These events, along with the Office.MATS namespace events, are used for the following purposes:

1) Identify whether clients can successfully obtain an authentication token or have entered a broken authentication state.

2) Evaluate whether changes that have occurred on the client or services have resulted in critical regressions in the user's authentication experience and reliability

3) When failures occur, these signals emit important failure codes from the responsible component (Office client code, authentication libraries, or authority services) which can be used for triage, diagnosis and mitigation

4) These signals power various ship readiness and health monitors, which fire alerts so our engineers can engage quickly and reduce the time to mitigation of critical user-blocking failures

The following fields are collected:

- **AllowSavedCreds** - whether the new credential is persisted
- **AuthScheme** - the authentication scheme used
- **CredsSaved** - whether the new credential is saved
- **DocumentUrlHash** - the encrypted URL requesting
- **EndTag** - the tag where the prompt ended
- **NewIdentity**_ErrorState – whether the new identity is valid
- **NewIdentity_HashedUniqueId** - the encrypted new identity ID after prompt is completed
- **NewIdentity_ProviderType** - the new identity provider after the prompt is completed
- **NewIdentity_UniqueID** - the new identity ID after the prompt is completed
- **OutStatus** - whether the prompt output is valid
- **PromptEndTime** - the time when the prompt ended
- **PromptFailedTag** - the tag that indicates SSPI prompt failure
- **PromptFlow** - the tag that invoked the SSPI prompt
- **PromptStartTime** - the time when the prompt started
- **Proxy** - if proxy is used
- **ServerHash** - the encrypted server address

- **SessionEndTime** - the time when the event session ended
- **Timeout** - the time when the prompt is timed out
- **UiMessage** - the UI message in the prompt
- **UserNameHash** - the encrypted user name

Office.Identity.Win32Prompt

Collected when Office shows the user a multi-factor authentication sign-in prompt. Along with silent token acquisitions, authentication prompts determine whether a user is in a broken authentication state, which results in an Offline Client state. Broken authentication may prevent license acquisition and result in a completely unusable client.

These events, along with the Office.MATS namespace events, are used for the following purposes:

1) Identify whether clients can successfully obtain an authentication token or have entered in to a broken authentication state.

2) Evaluate whether changes that have occurred on the client or services have resulted in critical regressions in the user's authentication experience and reliability

3) When failures occur, these signals emit important failure codes from the responsible component (Office client code, authentication libraries, or authority services) which can be used for triage, diagnosis and mitigation

4) These signals power various ship readiness and health monitors, which fire alerts so our engineers can engage quickly and reduce the time to mitigation of critical user-blocking failures.

The following fields are collected:

- **AdalWAMUsed** – the tag that indicates result if ADAL-atop-WAM is used
- **CallTag** – the tag that indicates the caller of sign in UI
- **Context** - the sign in context for the prompt
- **EndTagIdentityProviderRequested** - the tag where the identity provider is requested
- **HrdShownTag** - the tag where the HRD sign in dialog is shown
- **IdentityProviderResulted** - the identity provider type it is requesting
- **IdPFlowTag** - the tag that indicates the identity request result
- **LastLoginDelta** - the time delta from last successful login
- **NewIdentity_ErrorState** - whether the identity is valid after prompt
- **NewIdentity_ProviderType** - the new identity provider type after prompt
- **NewIdentity_UniqueID** - the new identity ID returned after prompt
- **PromptCorrelation** - the prompt correlation ID for diagnostic purpose
- **PromptEndTime** - the time when the prompt ended
- **PromptStartTime** – the time when the prompt started
- **SessionEndTime** - the time when the event session ended
- **ShowUIResult** – the result code returned from the prompt UI
- **StartTag** – the tag where Win32 prompt started

- **Timeout** - the time when the prompt is timed out
- **WasIdentitySignedOut** - whether the user is signed out

Office.MATS.actionofficewin32, Office.MATS.actionofficewinrt

The following description applies for both Win32 and WinRT events (the name depends on platform.)

Microsoft Auth Telemetry System (MATS) is collected when Office attempts to acquire an authentication token, either silently or through prompting. When acquisition attempts fail, error information is included. These events help our users avoid entering broken authentication states by:

1) Identifying whether clients can successfully obtain an authentication token or have entered a broken authentication state.

2) Evaluate when changes occur on the client or services, whether they result in critical regressions in the user's authentication experience and reliability

3) When failures occur, these signals emit important failure codes from the responsible component (Office client code, authentication libraries, or authority services) which can be used for triage, diagnosis and mitigation

4) These signals power various ship readiness and health monitors, which fire alerts so our engineers can engage quickly and reduce the time to mitigation of critical failures.

The following fields are collected:

- **Actiontype** - Which authentication library is used
- **Appaudience** - Is the app build for internal or external use
- **Appforcedprompt** - Did the app override cache and force a prompt to be shown
- **Appname** - Name of the application doing authentication
- **Appver** - Version of the application doing authentication
- **Askedforcreds** - Did the application ask the user to enter credentials for this action
- **Authoutcome** - Did the authentication attempt succeed, fail, or was canceled
- **Blockingprompt** - Did the application throw a prompt requiring user interaction
- **Correlationid** - GUID used for joining with services data
- **Count** - Count of events in cases of aggregation
- **Data_accounttype** - Consumer or organizational account
- **Devicenetworkstate** - Was the user online
- **Deviceprofiletelemetryid** - Anonymous device ID used to measure device experience
- **Duration** - How long the authentication took
- **Duration_Max** - If this signal is aggregated, the maximum duration of any aggregated event.
- **Duration_Min** - If this signal is aggregated, the minimum duration of any aggregated event.
- **Duration_Sum** - If this signal is aggregated, the sum of the durations of all the aggregated events.
- **Endtime** - When the authentication event ended
- **Error** - Error code if the authentication failed
- **Errordescription** - Brief description of the error

- **Errorsource** - Did the error come from service, authentication library, or application
- **Identityservice** - Was Microsoft Service Account (MSA) or Azure Active Directory (AAD) service invoked
- **Interactiveauthcontainer** - What type of prompt was shown
- **Issilent** - Was a prompt shown
- **Microsoft_ADAL_adal_version** - Version of the Azure Active Directory Authentication Library (ADAL)
- **Microsoft_ADAL_api_error_code** - Error code emitted by authentication library for this authentication attempt
- **Microsoft_ADAL_api_id** - API invoked for this authentication attempt
- **Microsoft_ADAL_authority** – Azure Active Directory authority URL responsible for authenticating the user
- **Microsoft_ADAL_authority_type** – Consumer / Microsoft Service Agreement (MSA) vs organizational / Azure Active Directory (AAD); currently always AAD
- **Microsoft_ADAL_authority_validation_status** – Tells whether authentication completed on the service-side
- **Microsoft_ADAL_broker_app** - Tells whether ADAL used a broker for authentication
- **Microsoft_ADAL_broker_app_used** – Tells the name of the broker (for example, Windows Account Management)
- **Microsoft_ADAL_broker_version** - Tells the version of the broker if used
- **Microsoft_ADAL_cache_event_count** - Number of cache events ADAL performed while retrieving token
- **Microsoft_ADAL_cache_event_count_max** - If this signal is aggregated, max cache events of any one of the aggregated events.
- **Microsoft_ADAL_cache_event_count_min** - If this signal is aggregated, min cache events of any one of the aggregated events.
- **Microsoft_ADAL_cache_event_count_sum** - If this signal is aggregated, sum of the cache events of all the aggregated events.
- **Microsoft_ADAL_cache_read_count** - How many times the API read from the disk cache. Present if there was at least one read.
- **Microsoft_ADAL_cache_read_error_count** - How many times the disk cache read failed. Is present if there was at least one failure.
- **Microsoft_ADAL_cache_read_last_error** - ADAL error code. Present if there was at least one read failure.
- **Microsoft_ADAL_cache_read_last_system_error** - System error code. Is present if there was at least one read failure.
- **Microsoft_ADAL_cache_write_count** - How many times the API wrote to the disk cache. Present if there was at least one write.
- **Microsoft_ADAL_cache_write_error_count** - How many times the disk cache-write failed. Present if there was at least one failure.
- **Microsoft_ADAL_cache_write_last_error** - ADAL error code. Present if there was at least one write

failure.

- **Microsoft_ADAL_cache_write_last_system_error** - System error code. Present if there was at least one write failure.
- **Microsoft_ADAL_client_id** - Hashed AAD app ID
- **Microsoft_ADAL_extended_expires_on_setting** - True/false telling if the token has an extended lifetime.
- **Microsoft_ADAL_http_event_count** - Count of HTTP calls made by ADAL.
- **Microsoft_ADAL_http_event_count_max** - If this signal is aggregated, max HTTP calls made by ADAL of any aggregated event.
- **Microsoft_ADAL_http_event_count_min** - If this signal is aggregated, min HTTP calls made by ADAL of any aggregated event.
- **Microsoft_ADAL_http_event_count_sum** - If this signal is aggregated, sum of the HTTP calls made by ADAL of all the aggregated events.
- **Microsoft_ADAL_is_silent_ui** - True/false telling if UI was shown (prompt) by ADAL.
- **Microsoft_ADAL_is_successful** - True/false telling if ADAL API succeeded.
- **Microsoft_ADAL_logging_pii_enabled** - True/false telling if ADAL full logging mode is enabled. This data is only logged locally, not emitted in telemetry.
- **Microsoft_ADAL_oauth_error_code** – OAuth protocol error code returned by the service.
- **Microsoft_ADAL_prompt_behavior** - log-in or none HTTP parameter passed to service to specify if user interface can be shown.
- **Microsoft_ADAL_request_id** - Transactional GUID for the request emitted by ADAL to the service.
- **Microsoft_ADAL_response_code** - HTTP response code from the service.
- **Microsoft_ADAL_response_time** - How long it took service to return to ADAL.
- **Microsoft_ADAL_response_time_max** - If the signal is aggregated, the max time it took ADAL to return from its API among any of the aggregated events.
- **Microsoft_ADAL_response_time_min** - If the signal is aggregated, the min time it took the service to respond to ADAL among any of the aggregated events.
- **Microsoft_ADAL_response_time_sum** - If the signal is aggregated, the sum of the time it took ADAL to return from its API among all aggregated events.
- **Microsoft_ADAL_rt_age** – Age of the refresh token
- **Microsoft_ADAL_server_error_code** - Error code returned by the server
- **Microsoft_ADAL_server_sub_error_code** - Sub error code returned by the server to help disambiguate why the request failed.
- **Microsoft_ADAL_spe_ring** - True/false telling if the user was using the Secure Production Enterprise inner ring (Microsoft employees only).
- **Microsoft_ADAL_start_time** – Time the ADAL API call was made
- **Microsoft_ADAL_stop_time** - Time the ADAL API call returned
- **Microsoft_ADAL_telemetry_pii_enabled** - True/false telling if ADAL full telemetry mode is enabled.

The name is a misnomer, as no PII/EUII is emitted.

- **Microsoft_ADAL_tenant_id** - GUID identifying the tenant that the authenticated user belongs to.
- **Microsoft_ADAL_token_acquisition_from_context** - Describes the ADAL behavior based on the tokens in the authentication context.
- **Microsoft_ADAL_token_type** – Either refresh token (RT) or multi-resource refresh token (MRRT).
- **Microsoft_ADAL_ui_event_count** - Count of prompts shown to the user. May have been silent.
- **Microsoft_ADAL_user_cancel** - True / false if the user interface window was canceled.
- **Microsoft_ADAL_was_request_throttled** – True / false indicating if this event was throttled by ADAL due to too many requests.
- **Microsoft_ADAL_x_ms_request_id** – Additional request ID provided in HTTP header to service by ADAL.
- **Platform** - Win32/WinRT/Android/iOS/Mac
- **Promptreasoncorrelationid** – For prompts, this is the correlation id of another event which explains why the user might be seeing an authentication prompt.
- **Resource** – The resource that the user is requesting a token for, such as Exchange or SharePoint.
- **Scenarioid** – GUID. Multiple events may belong to a single scenario, for example, the scenario may be adding a new account but there are multiple prompts that occur as part of that scenario. This ID enables correlation to happen.
- **Scenarioname** – The name of the scenario that this authentication event belongs to.
- **Sessionid** - GUID identifying the boot session
- **Skdver** - Version of MATS client SDK used to produce this data
- **Starttime** - Time at which the Start*Action MATS API was called
- **Tenantid** - GUID identifying the tenant the authenticated user belongs to (in non-ADAL cases).
- **Uploadid** - Unique GUID for this event, used for de-duping
- **Wamapi** - Identifies which WAM API is called
- **Wamtelemetrybatch** - Currently unused. In the future, allows the WAM component to dispatch additional information regarding the authentication event.

Office.MATS.OneAuth.ActionMicrosoftOfficeWin32

Microsoft Auth Telemetry System (MATS) is collected when Office attempts to acquire an authentication token, either silently or through prompting. When acquisition attempts fail, error information is included. These events help our users avoid entering broken authentication states by:

1. Identifying whether clients can successfully obtain an authentication token from the service, or have entered a broken authentication state.
2. Evaluate when changes occur on the client or services, whether they result in critical regressions in the user's authentication experience and reliability
3. When failures occur, these signals emit important failure codes from the responsible component (Office client code, authentication libraries, or authority services) which can be used for triage, diagnosis and mitigation

4. These signals power various ship readiness and health monitors, which fire alerts so our engineers can engage quickly and reduce the time to mitigation of critical failures.

The following fields are collected:

- **Accounttype** - Type of the account used for this authentication event, for example, consumer or organizational.
- **Actionname** - Friendly name for this event, if one was provided.
- **Actiontype** - Specifies the type of authentication library in use.
- **Appaudience** - Is the app build for internal or external use
- **Appforcedprompt** - Did the app override cache and force a prompt to be shown
- **Appname** - Name of the application doing authentication
- **Appver** - Version of the application doing authentication
- **Askedforcreds** - Did the application ask the user to enter credentials for this action
- **Authoutcome** - Did the authentication attempt succeed, fail, or was canceled
- **Blockingprompt** - Did the application throw a prompt requiring user interaction
- **Correlationid** - Identifier used to join information regarding this individual event with services data
- **Count** - The total number of aggregated actions reported in this one data event.
- **Devicenetworkstate** - Is the device connected to the internet.
- **Deviceprofiletelemetryid** - Anonymous device ID used to measure device-wide authentication experience and reliability.
- **Duration** - How long the authentication took
- **duration_max** - Max duration of any one of the aggregated events
- **duration_min** - Min duration of any one of the aggregated events
- **duration_sum** - Sum of the duration of all the aggregated events
- **endtime** - When the authentication event ended
- **error** - Error code if the authentication failed
- **errordescription** - Brief description of the error
- **errorsource** - Did the error come from service, authentication library, or application
- **eventtype** - Is this event reporting an authentication datapoint, or a data quality error event. Used to measure data quality.
- **from_cache** - Boolean representing whether the record is from the WAM core cache, or the plugin
- **hasadaltelemetry** - Indicates whether the Azure Active Directory Authentication Library (ADAL) provided telemetry for this event.
- **Identityservice** - Was Microsoft Service Account (MSA) or Azure Active Directory (AAD) service invoked
- **Interactiveauthcontainer** - What type of prompt was shown
- **Issilent** - Was a prompt shown or was this a silent (background) authentication event.

- **Microsoft_ADAL_adal_version** - Version of the Azure Active Directory Authentication Library (ADAL)
- **Microsoft_ADAL_api_error_code** - Error code emitted by authentication library for this authentication attempt
- **Microsoft_ADAL_api_id** - API invoked for this authentication attempt
- **Microsoft_ADAL_application_name** - The name of the application / process using ADAL.
- **Microsoft_ADAL_application_version** - The version of the application using ADAL.
- **Microsoft_ADAL_authority** - Azure Active Directory authority URL responsible for authenticating the user
- **Microsoft_ADAL_authority_type** - Consumer / Microsoft Service Agreement (MSA) vs organizational / Azure Active Directory (AAD); currently always AAD
- **Microsoft_ADAL_authority_validation_status** - Tells whether authentication completed on the service-side
- **Microsoft_ADAL_broker_app** - Tells whether ADAL used a broker for authentication
- **Microsoft_ADAL_broker_app_used** - Tells the name of the broker (for example, Windows Account Management)
- **Microsoft_ADAL_broker_version** - Tells the version of the broker if used
- **Microsoft_ADAL_cache_event_count** - Number of cache events ADAL performed while retrieving token
- **Microsoft_ADAL_cache_event_count_max** - If this signal is aggregated, max cache events of any one of the aggregated events
- **Microsoft_ADAL_cache_event_count_min** - If this signal is aggregated, min cache events of any one of the aggregated events
- **Microsoft_ADAL_cache_event_count_sum** - If this signal is aggregated, sum of the cache events of all the aggregated events
- **Microsoft_ADAL_cache_read_count** - How many times the API read from the disk cache. Present if there was at least one read
- **Microsoft_ADAL_cache_read_error_count** - How many times the disk cache read failed. Is present if there was at least one failure
- **Microsoft_ADAL_cache_read_last_error** - ADAL error code. Present if there was at least one read failure
- **Microsoft_ADAL_cache_read_last_system_error** - System error code. Is present if there was at least one read failure
- **Microsoft_ADAL_cache_write_count** - How many times the API wrote to the disk cache. Present if there was at least one write
- **Microsoft_ADAL_cache_write_error_count** - How many times the disk cache-write failed. Present if there was at least one failure
- **Microsoft_ADAL_cache_write_last_error** - ADAL error code. Present if there was at least one write failure
- **Microsoft_ADAL_cache_write_last_system_error** - System error code. Present if there was at least one write failure

- **Microsoft_ADAL_client_id** - Hashed Azure Active Directory app ID
- **Microsoft_ADAL_device_id** - ADAL-generated local device id.
- **Microsoft_ADAL_error_domain** - The domain/component that generated the error code.
- **Microsoft_ADAL_error_protocol_code** - OAuth protocol error code returned by the service, recorded by ADAL.
- **Microsoft_ADAL_extended_expires_on_setting** - True/false telling if the token has an extended lifetime
- **Microsoft_ADAL_http_event_count** - Number of HTTP requests generated by ADAL.
- **Microsoft_ADAL_idp** - The Identity Provider (idp) used by ADAL.
- **Microsoft_ADAL_network_event_count** - Count of network calls made by ADAL
- **Microsoft_ADAL_http_event_count_max** - If this signal is aggregated, max of http calls made by ADAL
- **Microsoft_ADAL_http_event_count_min** - If this signal is aggregated, min of http calls made by ADAL
- **Microsoft_ADAL_http_event_count_sum** - If this signal is aggregated, sum of http calls made by ADAL
- **Microsoft_ADAL_network_event_count_max** - If this signal is aggregated, max network calls made by ADAL of any aggregated event
- **Microsoft_ADAL_network_event_count_min** - If this signal is aggregated, min network calls made by ADAL of any aggregated event
- **Microsoft_ADAL_network_event_count_sum** - If this signal is aggregated, sum of the network calls made by ADAL of all the aggregated events
- **Microsoft_ADAL_is_silent_ui** - True/false telling if UI was shown (prompt) by ADAL
- **Microsoft_ADAL_is_successfull** - True/false telling if ADAL API succeeded (macOS)
- **Microsoft_ADAL_is_successful** - True/false telling if ADAL API succeeded
- **Microsoft_ADAL_logging_pii_enabled** - True/false telling if ADAL full logging mode is enabled. This data is only logged locally, not emitted in telemetry
- **Microsoft_ADAL_ntlm** - True/false telling if ADAL used basic auth (NTLM).
- **Microsoft_ADAL_oauth_error_code** - OAuth protocol error code returned by the service
- **Microsoft_ADAL_prompt_behavior** - log-in or none network parameter passed to service to specify if user interface can be shown
- **Microsoft_ADAL_request_id** - Transactional GUID for the request emitted by ADAL to the service
- **Microsoft_ADAL_response_code** - network response code from the service
- **Microsoft_ADAL_response_time** - How long it took service to return to ADAL
- **Microsoft_ADAL_response_time_max** - If the signal is aggregated, the max time it took ADAL to return from its API among any of the aggregated events
- **Microsoft_ADAL_response_time_min** - If the signal is aggregated, the min time it took the service to respond to ADAL among any of the aggregated events

- **Microsoft_ADAL_response_time_sum** - If the signal is aggregated, the sum of the time it took ADAL to return from its API among all aggregated events
- **Microsoft_ADAL_rt_age** - Age of the refresh token
- **Microsoft_ADAL_server_error_code** - Error code returned by the server
- **Microsoft_ADAL_server_sub_error_code** - Sub error code returned by the server to help disambiguate why the request failed
- **Microsoft_ADAL_spe_info** - True/false telling if the user was using the Secure Production Enterprise inner ring (Microsoft employees only)
- **Microsoft_ADAL_spe_ring** - True/false telling if the user was using the Secure Production Enterprise inner ring (Microsoft employees only)
- **Microsoft_ADAL_start_time** - Time the ADAL API call was made
- **Microsoft_ADAL_status** - Success/Failure status on the overall ADAL invocation
- **Microsoft_ADAL_stop_time** - Time the ADAL API call returned
- **Microsoft_ADAL_telemetry_pii_enabled** - True/false telling if ADAL full telemetry mode is enabled. The name is a misnomer, as no PII/EUII is emitted
- **Microsoft_ADAL_tenant_id** - GUID identifying the tenant that the authenticated user belongs to
- **Microsoft_ADAL_token_acquisition_from_context** - Describes the ADAL behavior based on the tokens in the authentication context
- **Microsoft_ADAL_token_frt_status** - Status of the refresh token: whether it was tried, not needed, not found, or deleted.
- **Microsoft_ADAL_token_mrrt_status** - Status of the MultiResourceRefreshToken: whether it was tried, not needed, not found, or deleted.
- **Microsoft_ADAL_token_rt_status** - Status of the refresh token: whether it was tried, not needed, not found, or deleted.
- **Microsoft_ADAL_token_type** - Either refresh token (RT) or multi-resource refresh token (MRRT)
- **Microsoft_ADAL_ui_event_count** - Count of prompts shown to the user. May have been silent
- **Microsoft_ADAL_user_cancel** - True / false if the user interface window was canceled
- **Microsoft_ADAL_x_ms_request_id** - Additional request ID provided in network header to service by ADAL
- **Microsoft_ADAL_x_client_cpu** - Information regarding the CPU Architecture of the device
- **Microsoft_ADAL_x_client_os** - The device OS Version.
- **Microsoft_ADAL_x_client_sku** - The name of the device OS SKU.
- **Microsoft_ADAL_x_client_ver** - The version of the ADAL library.
- **MSAL_all_error_tags** - All error tags the Microsoft Authentication Library (MSAL) encountered during the authentication flow.
- **MSAL_api_error_code** - If MSAL encounters an error bubbled up from the OS, platform error codes are stored here.
- **MSAL_api_error_context** - String containing additional human readable details about the last error

MSAL encountered.

- **MSAL_api_error_tag** - Unique string for the place in code where this error occurred.
- **MSAL_api_name** - Name of the MSAL top-level API called to start this authentication flow.
- **MSAL_api_status_code** - Status code MSAL returned for this authentication flow result.
- **MSAL_auth_flow** - Steps MSAL attempted during this authentication flow (AT, PRT, LRT, FRT, ART, IRT). Separated by the pipe "|" symbol for easy parsing.
- **MSAL_auth_flow_last_error** - Error code we received from the server on the second to last item in AuthFlow. (Ex: if AuthFlow = "PRT|LRT", PRT's error would be in AuthFlowLastError).
- **MSAL_authority_type** - Was this request for a user in: AAD, Federated, or MSA.
- **MSAL_broker_app_used** - Was a broker app used in this auth flow.
- **MSAL_client_id** - Client ID of the calling application
- **MSAL_correlation_id** - Unique GUID for this event, used to join actions across client, server, and app logs.
- **MSAL_delete_token** - List of tokens that were deleted from cache during this authentication flow.
- **MSAL_http_call_count** - Number of HTTP calls MSAL made during the authentication flow.
- **MSAL_is_successful** - Was the authentication flow successful.
- **MSAL_last_http_response_code** - If MSAL made one or more HTTP call(s), this is the last HTTP response code we received.
- **MSAL_msal_version** - MSAL's version string, format X.X.X+("OneAuth", "local", or a commit hash).
- **MSAL_read_token** - Tokens that were read from cache (AT, ART, FRT, LRT, IRT, PRT, EAT [EAT = Expired AT was read, but discarded]).
- **MSAL_read_token_last_error** - If MSAL encountered an error reading from cache, we'll store info here. (Ex: Disk read error bubbled from OS, Keychain error on macOS).
- **MSAL_request_duration** - How long the request took from when MSAL's top-level API was called, until we returned a result.
- **MSAL_request_id** - Request ID for the last call we made to Microsoft's secure token service.
- **MSAL_server_error_code** - Microsoft specific secure token service numeric error code if we received one.
- **MSAL_server_spe_ring** - Microsoft secure token service's Secure Production Enterprise ring info if we received it.
- **MSAL_server_suberror_code** - Microsoft specific secure token service suberror code string if we received one.
- **MSAL_start_time** - Time MSAL request was started at the top-level public API.
- **MSAL_stop_time** - Time MSAL finished processing the request and returned a result to the caller.
- **MSAL_tenant_id** - Microsoft GUID identifying the tenant the user exists in.
- **MSAL_ui_event_count** - Number of UI prompts MSAL displayed on screen.
- **MSAL_wam_telemetry** - Contains a batch of WAM telemetry data in a JSON string that will be parsed

and converted to the fields in this document that are sourced from WAM.

- **MSAL_was_request_throttled** - True if MSAL throttled this request and prevented it from hitting network. If this is ever true, there is most likely a loop in the calling app.
- **MSAL_write_token** - Tokens that were written to cache (AT, ART, FRT, LRT, IRT, PRT, EAT [EAT = Expired AT was read, but discarded]).
- **MSAL_write_token_last_error** - If MSAL encountered an error writing to cache, we'll store info here. (Ex: Disk read error bubbled from OS, Keychain error on macOS).
- **oneauth_api** - OneAuth API invoked for this authentication attempt.
- **oneauth_transactionuploadid** - GUID specifying an individual call to the OneAuth API.
- **oneauth_version** - The version of the OneAuth SDK.
- **Platform** - OS Platform (0: Windows Desktop, 1: Android, 2: iOS, 3: macOS, 4: UWP)
- **Promptreasoncorrelationid** - A correlation identifier that can be used to look up a previous authentication event, which is used to explain why the user was prompted to authenticate.
- **Resource** - The resource for which a token is requested.
- **Scenarioid** - Multiple events may belong to a single scenario, for example, the scenario may be adding a new account but there are multiple prompts that occur as part of that scenario. This identifier enables correlation of those related events.
- **Scenarioname** - Name of the application scenario where authentication was required, for example, first-boot, licensing check, etc.
- **Scope** - The scope for which a token is requested.
- **Sdkver** - Version of Microsoft Auth Telemetry System library used to produce this data
- **Sessionid** - Identifier for the boot session
- **Starttime** - Time at which the authentication event began.
- **Tenantid** - GUID identifying the tenant the authenticated user belongs to (in non-ADAL cases)
- **Uploadid** - Unique GUID for this event, used for de-duping
- **wamapi** - Identifies which Windows Web Account Management (WAM) API is called
- **wamtelemetrybatch** - Currently unused. In the future, allows the WAM component to dispatch additional information regarding the authentication event
- **WAM_account_join_on_end** - Account join state at the end of a WAM operation. Possible values: “primary”, “secondary”, “not_joined”
- **WAM_account_join_on_start** - Account join state at the start of a WAM operation. Possible values: “primary”, “secondary”, “not_joined”
- **WAM_api_error_code** - If an error response came from the AAD WAM plugin, this field will exist and will contain that error code
- **WAM_authority** - String containing the authority url—this should be the login.windows.net endpoint used
- **WAM_broker_version** - Present if WAM was used, this is the broker version string
- **WAM_cache_event_count** - The number of WAM cache events within the operation

- **WAM_client_id** - Identifier for joining with services data, this identifies the client application.
- **WAM_correlation_id** - Identifier for joining events with services data
- **WAM_device_join** - The device join state; possible values are "aadj", "haadj"
- **WAM_network_event_count** - Present if at least one network call happened; the number of network calls to the service for that WAM operation
- **WAM_network_status** - Present if at least one network call happened, contains an HTTP error code if the network request failed.
- **WAM_idp** - Specifies if the WAM consumer or organizational auth plugin was used.
- **WAM_is_cached** - Specifies if the response provided by WAM was retrieved from cache.
- **WAM_oauth_error_code** - Contains the error code returned by the service as part of the oauth protocol.
- **WAM_prompt_behavior** - Specifies if this prompt is forced by the app, or, if this request might skip prompting if it can silently authenticate.
- **WAM_provider_id** - Specifies the Microsoft endpoint for the authority in use for the auth scenario.
- **WAM_redirect_uri** - The redirect URI registered for the application in Azure Active Directory.
- **WAM_resource** - The resource for which a token is requested.
- **WAM_server_error_code** - The error code returned by the service to WAM.
- **WAM_server_sub_code** - An additional error code used to further break down the causes for failure, returned by the service.
- **WAM_silent_code** - The error code encountered by the internal silent attempt WAM makes, prior to prompting the user.
- **WAM_silent_mats** - Unused.
- **WAM_silent_message** - The error message associated with the internal silent attempt WAM makes, prior to prompting the user.
- **WAM_silent_status** - The success/fail status for the internal silent attempt WAM makes, prior to prompting the user.
- **WAM_tenant_id** - An identifier for the tenant the authenticated AAD user belongs to, if returned by the service
- **WAM_ui_visible** - Present if at least one UI window was shown to the user, either 'true' or 'false'
- **WAM_x_ms_clitelem** - Present if service returns header "x-ms-clitelem"

Office.MATS.OneAuth.TransactionMicrosoftOfficeWin32

Microsoft Auth Telemetry System (MATS) is collected when Office attempts to acquire an authentication token, either silently or through prompting. This event is a parent of one or more ActionMicrosoftOffice events, allowing related events to be grouped together. These events help our users avoid entering broken authentication states by:

1. Identifying whether clients can successfully obtain an authentication token from the service, or have entered a broken authentication state.
2. Evaluate when changes occur on the client or services, whether they result in critical regressions in the user's authentication experience and reliability

3. When failures occur, these signals emit important failure codes from the responsible component (Office client code, authentication libraries, or authority services) which can be used for triage, diagnosis and mitigation
4. These signals power various ship readiness and health monitors, which fire alerts so our engineers can engage quickly and reduce the time to mitigation of critical failures.

The following fields are collected:

- **Actiontype** - "oneauthtransaction" is the only value.
- **Appaudience** - Application audience (Automation, Preproduction, or Production)
- **Appname** - App name
- **Appver** - App version
- **Authoutcome** - Did the authentication attempt succeed, fail, or was canceled
- **Correlationid** - Identifier used to join information regarding this individual event with services data
- **Count** - Number of times the error occurred
- **Devicenetworkstate** - Device network state
- **Deviceprofiletelemetryid** - Device profile telemetry ID (string used by MATS to identify a specific device)
- **duration_max** - Minimum duration, in milliseconds, of the transactions aggregated on this signal.
- **duration_min** - Maximum duration, in milliseconds, of the transactions aggregated on this signal.
- **duration_sum** - Sum of durations, in milliseconds, of the transactions aggregated on this signal.
- **Endtime** - Time at which the OneAuth transaction ended.
- **Error** - OneAuth status code.
- **Eventtype** - Event type
- **Issilent** - False if UI was shown; true if it was a background event.
- **oneauth_Activeflights** - The list of flights that are active in the session, used for AB testing.
- **oneauth_api** - Specifies the public API of OneAuth that was invoked.
- **oneauth_Domain** - If the API call resulted in an error, this is the system domain of that error.
- **oneauth_ErrorCode** - Error code representing the internal error state for OneAuth. Replaces the old oneauth_errortag field.
- **oneauth_errortag** - Numerical identifier for a line of code that was responsible for generating an error.
- **oneauth_ExecutionFlow** - A series of tags identifying the codepath this API invocation took.
- **oneauth_internalerror** - Error code representing the internal error state for OneAuth.
- **oneauth_ServerErrorCode** - The server error returned to OneAuth at the conclusion of this API call, if one was encountered.
- **oneauth_SystemErrorCode** - The system error returned to OneAuth at the conclusion of this API call, if one was encountered.
- **oneauth_Tag** - The OneAuth tag designating the final place in code reached at the conclusion of this API

call.

- **oneauth_transactionuploadid** - Specifies the randomly generated internal GUID that maps to the specific invocation of a OneAuth API.
- **oneauth_version** - The version of the OneAuth SDK.
- **Platform** - OS Platform (0: Win32, 1: Android, 2: iOS, 3: macOS, 4: WinRT
- **Scenarioname** - Name of the scenario for which auth is necessary, specified by the calling application.
- **Schemaver** - Schema Version
- **Sdkver** - Version of the MATS sdk
- **Sessionid** - Session ID
- **severityError** - severity
- **starttime** - Time at which the OneAuth transaction began.
- **Timestamp** - Timestamp
- **Type** - Error type
- **Uploaded** - Unique identifier for this particular event, for de-duping purposes.

OneNote.SignIn.SSOExternalAppsAccountFound

This event is logged when an account with a valid refresh token is found among the list of accounts provided by TokenSharingManager. This scenario is specific to Single Sign-on (SSO).

The following fields are collected:

- **AccountType** - Logs the type of account
- **ProviderPackageID** - Logs the package ID of the app that provided this account

OneNote.SignIn.SSOExternalAppsInvalidAccount

This event is logged when there was an error when attempting to obtain a refresh token for an account in the list of accounts provided by TokenSharingManager. This scenario is specific to Single Sign-on (SSO)

The following fields are collected:

- **RawError** - Logs the raw error obtained when attempting to get a refresh token with the given account

OneNote.StickyNotes.FetchTokenCompleted

This event is logged post authentication, once fetching of refresh token is completed.

The following fields are collected:

- **ErrorMessage** - If fetching of token failed, this would log the error message
- **Result** - Logs the result of token fetching attempt
- **StickyNoteAccountType** - Logs type of the account for which the app was trying to fetch refresh token

Click-to-Run events

Office.ClickToRun.Bootstrapper

Office set up and inventory data collected when the user is running Office setup.exe to modify their installed Office products. Used to measure success / failure of a full user-initiated Office installation including pre-requisite checks.

The following fields are collected:

- **Data_BootStrapperStateFailure_ErrorCode** – The error code we failed with
- **Data_BootStrapperStateFailure_ErrorSource** – The function we failed in
- **Data_BootStrapperStateFailure_FailingState** – The part that we failed in the boostrapperbootstrapper
- **Data_BootStrapperStateFailure_OExceptionType** – The type of exception we failed with
- **Data_Culture** - the culture we are running this exe with, i.e. en-us
- **Data_HashedOLSToken** - a sha-256 hash of a token the OLS service gives us
- **Data_Platform** - x64 or x86 install
- **Data_PrereqFailure_Type** – The prerequisite failure we hit, i.e. the operating system is not supported
- **Data_ProductReleaseId** - Product we're installing, i.e. Microsoft 365 Apps for enterprise

Office.ClickToRun.CorruptionCheck

Office set up and inventory data collected when Click-to-Run client is running a corruption check to make sure that Office binaries are correct. Used to measure corruption of Office binaries, and which binaries are corrupt.

The following fields are collected:

- **Data_Active** - The current stream manifest we're checking on disk
- **Data_ActivePackages** - what packages the manifest contains
- **Data_ActiveVersion** - the version of the manifest
- **Data_AddFileCount** - how many files we're adding
- **Data_AddFileFiles** - a sample of the files we're adding
- **Data_CompressionLevel** - how the files are compressed
- **Data_CorruptionCheckLevel** - how deeply we're checking for corruption, stages
- **Data_CorruptSizeCount** - how many files have a corrupt size
- **Data_CorruptSizeFiles** - a sample of the files that have a corrupted size
- **Data_CorruptVersionCount** - how many files have a corrupted version
- **Data_CorruptVersionFiles** - a sample of the files that have a corrupted version
- **Data_FileBadDigestCount** -how many files we failed to open
- **Data_FileBadDigestFiles** - a sample of the files that we were unable to open
- **Data_FileNotSignedCount** - have many files that aren't signed
- **Data_FileNotSignedFiles** - a sample of the files that aren't signed
- **Data_FileNotTrustedCount** - how many files aren't trusted
- **Data_FileNotTrustedFiles** - a sample of the files that we don't trust
- **Data_IncompleteFileCount** - how many files seem to be incomplete
- **Data_IncompleteFileFiles** - a sample of the files that are incomplete

- **Data_KeepFileCount** - how many files we're not doing anything to
- **Data_KeepFileFiles** - a sample of files we're keeping
- **Data_KeepIncompleteFileCount** - how many files we're not changing despite them being incomplete
- **Data_KeepIncompleteFileFiles** - a sample of the files we're keeping that are incomplete
- **Data_MismatchSizeCount** - how many files have a size that doesn't match our manifest
- **Data_MismatchSizeFiles** - a sample of the files that are mismatched in size
- **Data_MismatchVersionCount** - how many files that have a version different than our manifest
- **Data_MismatchVersionFiles** - a sample of the files that have mismatched versions
- **Data_MissingFileCount** - how many files seem to be missing
- **Data_MissingFileFiles** - a sample of the files that are missing
- **Data_NotToBeStreamedFileCount** - how many files we're not streaming
- **Data_RemoveFileCount** - how many files we're removing
- **Data_RemoveFileFiles** - a sample of the files we're removing
- **Data_StreamUnitsMismatchCount** - how many files have units that don't match the manifest
- **Data_StreamUnitsMismatchFiles** - a sample of the files that have a stream with units mismatched
- **Data_TimeElapsed** - how long we took to check for corruption
- **Data_UpdateFileCount** - how many files we're updating
- **Data_UpdateFileFiles** - a sample of the files we're adding
- **Data_Working** - the new manifest we're checking
- **Data_WorkingVersion** - the version of the new manifest

Office.ClickToRun.MachineMetadata

Office set up and inventory data that provides necessary metadata for setup and inventory and is used to determine an accurate install base.

The following fields are collected:

- **Data_C2RClientVer** – The version of OfficeClickToRun.exe on the machine
- **Data_OfficeBitness** – The bitness that Office is installed in, x86 or x64
- **Data_OfficeVersion** - The version Office is installed in
- **Data_Sku** - The SKU that's installed, i.e. Microsoft 365 Apps for enterprise
- **Data_SqmMachineID** – Unique Machine ID used by Windows SQM Data_SusClientID- Machine Office update identifier

Office.ClickToRun.ODT

Office set up and inventory data collected when an IT Admin is running the Office Deployment Tool Click-to-Run setup.exe to modify their users' installed Office products. It is used to measure success / failure of full IT Admin initiated Office installations including pre-requisite checks.

The following fields are collected:

- **Data_BootStrapperStateFailure_ErrorCode**- The error code we failed with
- **Data_BootStrapperStateFailure_ErrorSource**- The function we failed in
- **Data_BootStrapperStateFailure_FailingState**- The part that we failed in the boot-strapper
- **Data_BootStrapperStateFailure_OExceptionType**- The type of exception we failed with
- **Data_ConfigurationHost**- The host where the configuration.xml came from
- **Data_ConfigurationId**- The ID we get from a configuration.xml
- **Data_ConfigurationSource**- We the configuration.xml came from
- **Data_Culture**- the culture we are running this exe with, i.e. en-us
- **Data_HashedOLSToken**- a sha-256 hash of a token the OLS service gives us
- **Data_MigrateArchRequest**- If we are migrating the user from x86 to x64 or vice-versa
- **Data_MigrateArchRequestValid**- If we believe the migrate request is valid
- **Data_Platform**- x64 or x86 install
- **Data_PlatformMigratedFrom**- Starting platform, i.e. x86
- **Data_PlatformMigratedTo**- Ending platform, i.e. x64
- **Data_PrereqFailure_Type**- The prerequisite failure we hit
- **Data_ProductReleaseId**- Product we're installing, i.e. Microsoft 365 Apps for enterprise

Office.ClickToRun.RepomanLogger

Reports on the status for the new Click-to-Run update pipeline ("Repoman") and if it successfully downloads and applies Office updates.

The following fields are collected:

- **ApplySucceeded** - True if the pipeline successfully applied an Office update, false if not.
- **DownloadSucceeded** - True if the pipeline successfully downloaded an Office update, false if not.
- **ErrorCode** - The code of the last error that occurred in the Click-to-Run Repoman pipeline.
- **ErrorDetails** - Additional error details of the last error that occurred in the Click-to-Run Repoman pipeline.
- **ErrorMessage** - The message of the last error that occurred in the Click-to-Run Repoman pipeline.
- **OpenStreamSessionSucceeded** - True if the pipeline successfully creates a session for streaming an Office update, false if not.
- **RepomanErrorMessage** - The error message received from the repoman.dll.

Office.ClickToRun.Scenario.InstallTaskConfigure

Office set up and inventory data collected when the Office installer is placing newly downloaded files. Used to measure the success / failure of an Office installation.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled

- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled** - If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts** - The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails** - Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove** - what Office products we're removing
- **Data_RemovingFixedProducts** - The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID** - Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.InstallTaskConfigurelight

Office set up and inventory data collected when the Office installer is deciding which files need to be downloaded. Used to measure success / failure of Office installation.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** -What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** -Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** -what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**-Machine Office update identifier

- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.InstallTaskFinalintegrate

Office set up and inventory data collected when the Office installer is installing licenses and registry settings. Used to measure success / failure of Office installation.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled** - If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts** - The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails** - Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove** - what Office products we're removing
- **Data_RemovingFixedProducts** - The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario

- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** -Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.InstallTaskFonts

Office set up and inventory data collected when the Office installer is installing fonts. Used to measure success / failure of Office installation.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** -Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine

- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** -what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.InstallTaskInitupdates

Office set up and inventory data collected when the Office installer is creating settings for updates to work properly. Used to measure success / failure of Office installation.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place

- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** –- Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.InstallTaskIntegrateinstall

Office set up and inventory data collected when the Office installer is creating registry entries for the Office applications Used to measure success / failure of Office installation.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add

- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.InstallTaskLastrun

Office set up and inventory data collected when Office installer is finishing the installation, pinning shortcuts and creating final registry settings. Used to measure success / failure of Office installation.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local

- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.InstallTaskMigrate

Office set up and inventory data collected when the Office Installer is migrating settings from older versions of Office. Used to measure success / failure of Office installation.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts** - The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** – - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI

- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.InstallTaskPublishrsod

Office set up and inventory data collected when the Office installer is publishing the virtual registry for the AppV virtualization layer. Used to measure success / failure of Office installation.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running, i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN

- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.InstallTaskRemoveinstallation

Office set up and inventory data collected when the Office uninstaller is removing parts of Office from the device. Used to measure success / failure of Office installation.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion**- What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing

- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.InstallTaskStream

Office set up and inventory data collected when the Office installer is downloading new files for Office. Used to measure success / failure of Office installation.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled** - If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts** - The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable

- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID** - Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.InstallTaskUninstallcentennial

Office set up and inventory data collected when the Office installer is uninstalling a previous version of Office installed from the Store. Used to measure success / failure of Office installation.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error

- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.RepairTaskFinalintegrate

Office set up and inventory data collected when the Office repair client republishes .msi files and Office extensions. Used to measure success / failure of Office repair.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion**- What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version

- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code can be ignored
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.RepairTaskFullrepair

Office set up and inventory data collected when the Office repair client downloads the latest version of the Click-to-Run client to prepare the computer for uninstall and reinstall. Used to measure success / failure of Office repair.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled

- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.RepairTaskIntegraterepair

Office set up and inventory data collected when the Office repair client attempts to repair some known troublesome registry entries. Used to measure success / failure of Office repair.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code can be ignored
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier

- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.RepairTaskRemoveinstallation

Office set up and inventory data collected when the Office repair client removes Office from the device to prepare for a reinstall when repairing. Used to measure success / failure of Office repair.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** -What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario

- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.UpdateTaskIntegrateupdate

Office set up and inventory data collected when the Click-to-Run client updates licenses if necessary. Used to measure success / failure of Office update.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine

- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.UpdateTaskPublishrsod

Office set up and inventory collected when the Click-to-Run client updates registry settings for new binaries. Used to measure success / failure of Office update.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place

- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.UpdateTaskUpdateapply

Office set up and inventory data collected when the Click-to-Run client shuts down running apps if needed and installs new files that were downloaded. Used to measure success / failure of Office update.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add

- **Data_AvailableVersion to**- What version of Office is available to update
- **Data_CompletedWithoutActionInfo** - Why we didn't complete the scenario, i.e. Apps were open
- **Data_CompletionState** - If we completed the task
- **Data_CorruptionChecksOnly** –- If we're only checking for corruption and not updating
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_HardlinkingException** - The exception we encountered when trying to create hardlinks
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_PackageOperationSuccessful** - True if we successfully completed our task on the Office package
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office
- **Data_WorkstationLockState** - True if we think the computer is locked

Office.ClickToRun.Scenario.UpdateTaskUpdateclientdownload

Office set up and inventory data collected when the Click-to-Run client downloads a newer version of itself. Used to measure success / failure of Office update.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName**- what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab

- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.UpdateTaskUpdatedetection

Office set up and inventory data collected when the Click-to-Run client checks if there is a new update available. Used to measure success / failure of Office update.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_AvailableVersion** - What version of Office is available to update to
- **Data_ComAction** - An int representing a com action we're performing
- **Data_CompletedWithoutActionInfo** - Why we didn't complete the scenario, i.e. Apps were open
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_PackageUpdateAvailable** - True if we have a new version of Office available
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing

- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.UpdateTaskUpdatedownload

Office set up and inventory data collected when the Click-to-Run client is downloading a new update. Used to measure success / failure of Office update.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion** - What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to
- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_AvailableVersion** - What version of Office is available to update to
- **Data_CompletedWithoutActionInfo**- Why we didn't complete the scenario, i.e. Apps were open
- **Data_CompletionState** - If we completed the task
- **Data_CorruptionChecksOnly** - If we're only checking for corruption and not updating
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place

- **Data_ExceptionType** - The exception we failed with
- **Data_FoundCorruptFiles** - True if we found corrupt files
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_PackageOperationSuccessful** - True if we successfully completed our task on the Office package
- **Data_PipelineExitCode** - The exit code our file pipeline returned
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID**- Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Scenario.UpdateTaskUpdatefinalize

Office set up and inventory data collected when the Click-to-Run Client is cleaning up from the update and restoring apps that were previously open. Used to measure success or failure of Office update.

The following fields are collected:

- **Data_15_SourceType** - Where the Office 15 source is located, i.e. CDN or Local
- **Data_15_UpdatesEnabled** - If Office 15 updates are enabled
- **Data_15_UpdateVersion**- What version of Office 15 we're updating to
- **Data_15_Version** - The Office 15 version
- **Data_16_SourceType** - Where the Office 16 source is located i.e. CDN or Local
- **Data_16_UpdatesEnabled**- If Office 16 updates are enabled
- **Data_16_UpdateVersion** - What version of Office 16 we're updating to

- **Data_16_Version** - The Office 16 version
- **Data_AddingFixedProducts**- The products we're adding
- **Data_AddingProducts** - What products we're asked to add
- **Data_CompletionState** - If we completed the task
- **Data_ErrorCode** - The error code we failed with
- **Data_ErrorDetails**- Extra details about an error
- **Data_ErrorMessage** - An error message on what went wrong
- **Data_ErrorSource** - Where the error took place
- **Data_ExceptionType** - The exception we failed with
- **Data_IsErrorCodeIgnorable** - If the error code we failed with is ignorable
- **Data_IsErrorCodeIgnorableForScenarioHealth** - If we believe the error code is ignorable
- **Data_NewestPackageVersion** - The newest version of Office on the machine
- **Data_OldestPackageVersion** - The oldest version of Office on the machine
- **Data_ProductsToAdd** - What Office products we're adding
- **Data_ProductsToRemove**- what Office products we're removing
- **Data_RemovingFixedProducts**- The products we're removing
- **Data_RemovingProducts** - The products we're asked to remove
- **Data_ScenarioInstanceID** - A unique GUID for the running scenario
- **Data_ScenarioName** - what scenario is running. i.e. install
- **Data_ScenarioSubType** - What type of scenario we're running, i.e. Uninstall, reinstall
- **Data_SourceType** - Where our source is, i.e. CDN
- **Data_SqmMachineID** - Unique Machine ID used by Windows SQM
- **Data_SusClientID** - Machine Office update identifier
- **Data_TaskState** - What state the task is in like running or canceled
- **Data_TotalClientCabSize** - The size of our client cab
- **Data_TriggeringUI** - What triggered the UI
- **Data_UpdatesEnabled** - If Office updates are enabled
- **Data_Version** - The version of Office

Office.ClickToRun.Transport

Reports on the file download actions to determine the success of the operation, type of download performed and diagnostic information.

- **BytesFromGroupPeers** - Bytes from group peers, only for downloads using Delivery Optimization
- **BytesFromHttp** - Bytes from http, only for downloads using Delivery Optimization
- **ByteFromInternetPeers** - Bytes from internet peers, only for downloads using Delivery Optimization

- **BytesFromLanPeers** - Bytes from Lan peers, only for downloads using Delivery Optimization
- **canceledJobs** - Number of canceled requests in the session
- **Connected** - Whether connected to source
- **ErrorCode** - The code of last error
- **ErrorDetails** - The detail of last error
- **ErrorMessage** - The message of last error
- **ErrorSource** - The source of last error, e.g. Connection, LoadFile or LoadRange
- **FailedJob** - Number of failed requests in the session
- **FileSize** - Size of the resource
- **SourcePathNoFilePath** - Source path of the resource only http source is reported, local file path or UNC path is filtered
- **SucceededJobs** - Number of succeeded requests in the session
- **TotalJobs** - Total number of requests in the session
- **TotalRequestedBytes** - Total requested bytes in the session
- **TotalTransferTime** - Total transfer time in the session
- **TransferredBytes** - Total transferred bytes in the session
- **TransportType** - Type of transport, e.g. (In memory Delivery Optimization, HTTP, Background Intelligent Transfer Service)

Office.ClickToRun.Transport.ExperimentalTransport.PipelineCreateTransport

Office set up and inventory data collected when the Click-to-Run client is creating a transport stream to download Office files. Used for determining the health of various transport technologies (e.g., HTTP, BITS, DO) which is critical to downloading Office properly for installation and updates.

The following fields are collected:

- **Data_IsForeGroundStreaming** – If we're streaming in the foreground or background
- **Data_IsInstallMode** – 1 if we're installing and downloading files, 0 if not
- **Data_SourceProtocol** – If we're downloading from a content data network, CDN, the machine we're installing on, local, or from a resource on the local area network,
- **Data_Status** – Success or failure

Office.ClickToRun.UpdateStatus

Office set up and inventory data collected when the Click-to-Run client is finishing an update status

The following fields are collected:

- **Data_build** - Currently installed Office version
- **Data_channel** – The channel that the user is on
- **Data_errorCode** – An integer code specifying the type of error that occurred, if there was one
- **Data_errorMessage** – A string giving a description of the error that occurred, if there was one
- **Data_status** – A short status of what happened during the update, such as Succeeded or Downloaded

- **Data_targetBuild** - -The Office version that we are attempting to update to

Office.ClickToRun.UniversalBootstrapper.Application

Reports the result of the end-to-end installation attempt

- **ErrorCode** – Integer value associated with an unhandled exception
- **ErrorDetails** – String that describes the location of where an unhandled exception occurred (function, file, line number, additional parameters set by the thrower)
- **ErrorMessage** – String defined at the point where an unhandled exception was thrown, describing the nature of the failure
- **ErrorType** – String describing the category of an unhandled exception
- **ExitCode** – Integer value associated with the result of running the bootstrapper, indicating success vs. specific types of failure

Office.ClickToRun.UniversalBootstrapper.CalculateParameters

Reports on the action that reason over the input collected using CollectParameters

- **BitField** – Integer value of the BitField argument, which tells us whether an explicit installation/update channel requested. For example, Beta Channel, Current Channel (Preview), Current Channel, Monthly Enterprise Channel, Semi-Annual Enterprise Channel (Preview), or Semi-Annual Enterprise Channel.
- **ChannelID** – Integer representing the enum value of the selected update/install channel. For example, Beta Channel, Current Channel (Preview), Current Channel, Monthly Enterprise Channel, Semi-Annual Enterprise Channel (Preview), Semi-Annual Enterprise Channel, or Invalid.
- **CMDMode** – The friendly string corresponding to which overall mode switch was detected in the cmd arguments passed to the exe.
- **C2RClientUICulture** – Culture of the C2R Client to install
- **ErrorCode** – Integer value associated with an unhandled exception
- **ErrorDetails** – String that describes the location of where an unhandled exception occurred (function, file, line number, additional parameters set by the thrower)
- **ErrorMessage** – String defined at the point where an unhandled exception was thrown, describing the nature of the failure
- **ErrorType** – String describing the category of an unhandled exception
- **ExcludedApps** – String listing the individual Office application names requested to be excluded from the Office suites installed
- **InstalledCabVersion** – The "16.0.xxxxx.yyyyy" version of an Office C2R Client already installed
- **InstalledProductVersion** – The "16.0.xxxxx.yyyyy" version of an Office C2R product already installed
- **IsC2RServiceRunning** – Boolean flag that indicates whether a modern C2R Client's local machine service is up and running on the device
- **IsElevatedFlagSet** – Boolean flag indicating whether the bootstrapper has already attempted to gain admin elevation
- **IsFireFlyInstalled** – Boolean flag indicating whether the Office 2013 RTM C2R Client is currently installed
- **IsFireflyServiceRunning** – Boolean flag that indicates whether a 2013 RTM C2R Client's local machine

service is up and running on the device

- **IsOfficeInstalled** – Boolean flag indicating whether a modern Office client is installed already
- **OfficeCultures** – Serialized list of Office cultures to be installed
- **OfficeSourceType** – Friendly string associated with the enum value of the installation source (CDN, HTTP, UNC, CMBITS, DVD, LOCAL)
- **Origin** – String value telling us which of the supported origins (Puerto Rico [PR], Singapore [SG], Dublin [DB]) should be used for initial installation streaming
- **PlatformFromLink** – String indicating the requested x86|x64|default bitness of Office requested from the C2R Setup service
- **PlatformOfExistingInstallation** – String indicating whether x86 vs. X64 Office was already installed on the device
- **PlatformToInstall** – String indicating the final decision on whether x86 vs. X64 Office should be installed. Possibilities are: autorun, configure, consumer, download, help, packager
- **PRID** – String value representing the requested Product Release ID in a consumer installation scenario (for example, "O365ProPlusRetail")
- **PridsToMigrateFromCentennial**- String of Office products to migrate from Store installations to Click-To-Run
- **ProductsToAdd** – The serialized string that instructs C2R Client on which Product/Culture combinations it should be installing
- **ProductsToMigrateFromO15C2R** - String of Office products and cultures to migrate from an Office 2013 Click-To-Run installation
- **ProductsToRemove** – The serialized string that instructs C2R Client on which Product/Culture combinations it should be uninstalling
- **SharedComputerLicensing** – Boolean indicating whether an IT Admin requested setup to enable the "SharedComputerLicensing" feature
- **ShouldActivate** – Boolean indicating whether an IT Admin requested an automatic licensing activation attempt in their configuration.xml
- **ShouldUninstallCentennial** - Boolean flag indicating whether Office products from the Store should be uninstalled
- **VersionToInstall** – String value of the Office "16.0.xxxxx.yyyyy" version that is being installed

Office.ClickToRun.UniversalBootstrapper.CollectEmbeddedSignature

Reports on the action that reads tagged input from the exe's embedded signature. This is an unproven concept the previous iteration of setup.exe did not implement, and is what we're relying on to carry the user's product/language/bitness choices from web page to in-process within setup.exe.

- **ErrorCode** – Integer number associated with an unhandled exception
- **ErrorDetails** – String that describes the location of where an unhandled exception occurred (function, file, line number, additional parameters set by the thrower)
- **ErrorMessage** – String defined at the point where an unhandled exception was thrown, describing the nature of the failure
- **ErrorType** – String describing the category of an unhandled exception

Office.ClickToRun.UniversalBootstrapper.CollectParameters

Reports the parameters used for the Office installation

- **BitField** – Integer value of the BitField argument, which tells us whether an explicit installation/update channel requested. For example, Beta Channel, Current Channel (Preview), Current Channel, Monthly Enterprise Channel, Semi-Annual Enterprise Channel (Preview), or Semi-Annual Enterprise Channel.
- **ChannelID** – Integer representing the enum value of the selected update/install channel. For example, Beta Channel, Current Channel (Preview), Current Channel, Monthly Enterprise Channel, Semi-Annual Enterprise Channel (Preview), Semi-Annual Enterprise Channel, or Invalid.
- **CMDMode** – The friendly string corresponding to which overall mode switch was detected in the cmd arguments passed to the exe. Possibilities are: autorun, configure, consumer, download, help, packager
- **C2RClientUICulture** – Culture of the C2R Client to install
- **ErrorCode** – Integer value associated with an unhandled exception
- **ErrorDetails** – String that describes the location of where an unhandled exception occurred (function, file, line number, additional parameters set by the thrower)
- **ErrorMessage** – String defined at the point where an unhandled exception was thrown, describing the nature of the failure
- **ErrorType** – String describing the category of an unhandled exception
- **ExcludedApps** – String listing the individual Office application names requested to be excluded from the Office suites installed
- **InstalledCabVersion** – The "16.0.xxxxx.yyyyy" version of an Office C2R Client already installed
- **InstalledProductVersion** – The "16.0.xxxxx.yyyyy" version of an Office C2R product already installed
- **IsC2RServiceRunning** – Boolean flag that indicates whether a modern C2R Client's local machine service is up and running on the device
- **IsElevatedFlagSet** – Boolean flag indicating whether the bootstrapper has already attempted to gain admin elevation
- **IsFireFlyInstalled** – Boolean flag indicating whether the Office 2013 RTM C2R Client is currently installed
- **IsFireflyServiceRunning** – Boolean flag that indicates whether a 2013 RTM C2R Client's local machine service is up and running on the device
- **IsOfficeInstalled** – Boolean flag indicating whether a modern Office client is installed already
- **OfficeCultures** – Serialized list of Office cultures to be installed
- **OfficeSourceType** – Friendly string associated with the enum value of the installation source (CDN, HTTP, UNC, CMBITS, DVD, LOCAL)
- **Origin** – String value telling us which of the supported origins (Puerto Rico [PR], Singapore [SG], Dublin [DB]) should be used for initial installation streaming
- **PlatformFromLink** – String indicating the requested x86|x64|default bitness of Office requested from the C2R Setup service
- **PlatformOfExistingInstallation** – String indicating whether x86 vs. X64 Office was already installed on the device

- **PlatformToInstall** – String indicating the final decision on whether x86 vs. X64 Office should be installed
- **PRID** – String value representing the requested Product Release ID in a consumer installation scenario (for example, "O365ProPlusRetail")
- **PridsToMigrateFromCentennial**- String of Office products to migrate from Store installations to Click-To-Run
- **ProductsToAdd** – The serialized string that instructs C2R Client on which Product/Culture combinations it should be installing
- **ProductsToMigrateFromO15C2R** - String of Office products and cultures to migrate from an Office 2013 Click-To-Run installation
- **ProductsToRemove** – The serialized string that instructs C2R Client on which Product/Culture combinations it should be uninstalling
- **SharedComputerLicensing** – Boolean indicating whether an IT Admin requested setup to enable the "SharedComputerLicensing" feature
- **ShouldActivate**– Boolean indicating whether an IT Admin requested an automatic licensing activation attempt in their configuration.xml
- **ShouldUninstallCentennial** - Boolean flag indicating whether Office products from the Store should be uninstalled
- **VersionToInstall**– String value of the Office "16.0.xxxxx.yyyyy" version that is being installed

Office.ClickToRun.UniversalBootstrapper.Execute

Reports on the machine-impactful actions taken, as determined by the reasoned-over data from "CalculateParameters"

- **AvailableClientVersionText** – String value of the C2R Client "16.0.xxxxx.yyyyy" version found in the Version Descriptor XML, which is used to determine whether a currently installed C2R Client should be updated
- **CleanFireflyAction** – "true" if the CleanFireFlyAction task is scheduled to run during this installation
- **CleanO15Action** – "true" if the CleanO15Action task is scheduled to run during this installation
- **CMDMode** – The friendly string corresponding to which overall mode switch was detected in the cmd arguments passed to the exe. Possibilities are: autorun, configure, consumer, download, help, packager
- **DeliveryMechanism** – The "FFNRoot" guid extracted from the Version Descriptor XML (stamped by RDX), which tells us which audience/channel the build source came from
- **DownloadC2RClientAction** – "true" if the DownloadC2RClientAction task is scheduled to run during this installation
- **ErrorCode** – Integer value associated with an unhandled exception
- **ErrorDetails** – String that describes the location of where an unhandled exception occurred (function, file, line number, additional parameters set by the thrower)
- **ErrorMessage** – String defined at the point where an unhandled exception was thrown, describing the nature of the failure
- **ErrorType** – String describing the category of an unhandled exception
- **ExitCode** – Integer value associated with the result of running the Execute phase of the bootstrapper,

indicating success vs. specific types of failure

- **LaunchAction** – "true" if the LaunchAction task is scheduled to run during this installation
- **LaunchUpdateAction** – "true" if the LaunchUpdateAction task is scheduled to run during this installation
- **PreReqResult** – The integer enum value of the result when PreReq checks were performed (pass/fail/re-run)
- **UnexpectedAction** – "true" if the UnexpectedAction task (an error case) is scheduled to run during this installation
- **VersionToInstall** – String value of the Office "16.0.xxxxx.yyyyy" version that is being installed

Office.ServiceabilityManager.InventoryAddon.Heartbeat

[This event has been removed from current builds of Office, but might still appear in older builds.]

This event is used to acquire standard metadata on each run of the Inventory add-on, which is part of the Office Serviceability Manager and is used for inventory Office information on those machines for which an IT admin has opted in. The metadata of specific interest here is the session ID, and it is used for linking to other data stored within a per-tenant cloud service.

This event contains no extra fields since only the metadata is relevant.

Office.ServiceabilityManager.InventoryAddon.Results

This event is logged when the call to the webservice made within the Click-to-Run Serviceability Manager Inventory add-on completes, irrespective of whether it succeeds or fails. This is essentially the last operation within the add-on to track the overall operation status.

The following fields are collected:

- **ActionDetail** - Additional details for when a failure occurs.
- **Result** - Numeric error code flags returned by the Office webservice call APIs. For example, 3 would mean that there was a problem initializing the HTTP headers.
- **Type** - Additional type information. In the case of the Inventory, this information specifies the type of payload being sent. For example, full or just a delta of changes.
- **Version** - The full four-part version number of Office. For example, 16.0.10000.10000. (Note that for these events, the standard version field is populated with the Windows version as this runs as part of a Windows process.)
- **WebCallSource** - An enumeration value (specified as an integer) indicating the Serviceability Manager add-on that was the source of the call.

Office.ServiceabilityManager.WebserviceFailure

Whenever a call to a webservice within one of the Office Serviceability Manager add-ins fails, this statement is logged. Failures can be due to internal failures or an inability to connect to the webservice.

The following fields are collected:

- **Add-on** - The Click-to-Run Serviceability Manager add-on from which the webservice call was made. This can have values like inventory, manageability, etc. encoded as a numeric value.
- **Correlation ID** - A randomly generated GUID specific to the current instance that is sent to the webservice to correlate calls between the client and the server.
- **ErrorInfo** - Numeric error code information returned by the Office webservice call APIs.

- **ErrorMessage** - A message providing further insight into the failure. Each error type maps to a hardcoded string, with some error types mapping to potentially multiple strings depending on the specific nature of the failure.
- **Function** - The function in the code from which the current call occurred.
- **Status** - The HTTP status code returned by the call to the webservice, e.g. 404, 500, etc.

Enhanced Configuration Service (ECS) events

Office.Experimentation.FeatureQueryBatched

Collects information about Feature gates/Change gates queried by executed code.

The following fields are collected:

- **Count** - Number of queried feature gates in this batched event
- **Features** - Information about the queried gate.
- **Sequence** - Order in which FeatureGate was queried

Office.Experimentation.FlightNumberLine

Collects the list of configurations received by the client from ECS

The following fields are collected:

- **ECSConfigs** - Comma-separated list of ECS Configs
- **LockType** - Type of FlightManager lock.
- **TasFlightingVersion** - Version number
- **TimeToLock** - Time between liblet initiation and FlightManager lock
- **UnmergedConfigs** - List of configurations not merged

Office.Experimentation.TriggerAnalysis

This event helps scope analysis of product usage and performance metrics (such as crashes, hangs, etc.) to the subset of users or devices that are eligible to use the feature, thereby helping ensure that the product is working properly.

The following fields are collected:

- **FeatureGate** - Identifies the set of features for which the trigger analysis is applicable.

OneNote.FlightDefault

This event is logged when OneNote asks ECS server for flight values. This is used to enable experimental features to those users who have opted in for receiving such flights.

The following fields are collected:

- **ConfigParam** - The config for which the value is being accessed

Licensing events

Office.Android.DocsUI.PaywallControl.AutoRedeemPendingPurchaseResult

Critical engineering telemetry to log the result of automatic attempt of trying to redeem pending purchases of a user. Product telemetry used for reconciliation of purchase transaction information with Microsoft's commerce system to enable associated subscription benefits.

The following fields are collected:

- **EventDate** – Timestamp of the event occurrence
- **Result** – Int denoting the enum result of the operation.
- **SessionID** – GUID to connect events by session

Office.Android.DocsUI.PaywallControl.PaywallUIShown

Critical Usage telemetry for when Paywall control is shown to the user. Used to understand the in app purchase experience for the user and optimize the same for future versions.

The following fields are collected:

- **EventDate** – Timestamp of the event occurrence
- **IsModeFRE** – Boolean to indicate experience type, Upsell dialog or SKU Chooser
- **SessionID** – GUID to connect events by session

Office.Android.DocsUI.PaywallControl.PurchaseButtonClicked

Critical Usage telemetry to know when user clicks on the Purchase Button. Used to infer the usage pattern and conversion metric for users who attempt to buy a subscription in the app.

The following fields are collected:

- **EventDate** – Timestamp of the event occurrence
- **IsDefaultSku** – Boolean indicating if user is attempting to purchase the Sku that showed up first/default
- **ProductID** – String identifying which subscription user is attempting to purchase as configured in the store
- **SessionID** – GUID to connect events by session

Office.Android.DocsUI.PaywallControl.PurchaseResult

Critical engineering telemetry to log the result of purchase attempt triggered manually by user. Product telemetry used for reconciliation of purchase transaction information with Microsoft's commerce system to enable associated subscription benefits.

The following fields are collected:

- **EventDate** – Timestamp of the event occurrence
- **IsModeFre** – Boolean indicating if purchase was made from upsell FRE screen or Sku Chooser
- **Result** – Int denoting the enum result of the operation.
- **SessionID** – GUID to connect events by session

Office.Android.DocsUI.PaywallControl.PurchaseTokenRedemptionResponse

[This event was previously named Office.Android.DocsUI.Views.PurchaseTokenRedemptionResponse.]

This product telemetry is collected for tracking and logging the internal transaction status and reconciliation information to improve reliability and performance. Microsoft uses this data to analyze and improve the reliability and performance of the internal transaction processing and reconciliation mechanisms.

The following fields are collected:

- **MicrosoftPurchaseOrderId** - Microsoft Order Id sent by Retail Federation Service (RFS) for tracking purposes.

- **ResponseCode** - HTTP Response code (int)
- **StatusCode** - RFS response status code (RFS defined Enum int- finite)

Office.Android.DocsUI.PaywallControl.SeeAllFeaturesAnalytics

We collect this usage telemetry to see how much time the user spends on the "See more benefits" screen. The data is used to understand usage of the "See more benefits" feature and further optimize the experience in future versions.

The following fields are collected:

- **Duration** - Long integer indicating time spent by user on "See All Features" screen in milliseconds
- **EventDate** - Timestamp of the event occurrence
- **MostExplored** - Integer denoting the index of the most toggled item in a list of Microsoft 365 apps and their features
- **SessionID** - Globally Unique Identifier (GUID) to connect events by session

Office.Android.DocsUI.PaywallControl.SkuChooserAnalytics

Usage telemetry to see how much time user spends on SKU Chooser screen. Usage telemetry to see how much time user spends on Sku Chooser screen.

The following fields are collected:

- **Duration** – Long integer indicating time spent by user on Sku Chooser screen in milliseconds
- **EventDate** – Timestamp of the event occurrence
- **SessionID** – GUID to connect events by session

Office.Android.DocsUI.PaywallControl.SkuPriceDiscountErrorEvent

The event is triggered when a user lands on the SKU chooser screen of the app and the prices are fetched from the Google Playstore for different subscriptions. The event identifies price differences between monthly and annual plan offered in different countries and in different currencies. The data is used to ensure that the pricing configuration is working as expected.

The following fields are collected:

- **CountryCode** - To identify the country where purchase is made.
- **Discount** - Discount Percentage offered based on price differences between monthly and annual SKU of the both the personal and family plans.
- **ProductIndex** - To identify whether personal or family plan.
- **StoreCurrencyCode** - To identify the currency in which the app store is offering the end users the subscriptions plans.

Office.Android.DocsUI.Views.DimeError

This event is collected for the Office app for Android (released on Huawei and in China Stores). This event indicates that an attempt to purchase a Microsoft 365 subscription through Dime (a webURL loaded in client webview) has failed. Only the error scenarios are captured. This event data is error data only and is used to ensure the health of the Dime purchase flow in the client.

The following fields are collected:

- **CorrelationID** - ID that uniquely identifies a Dime purchase session.
- **ErrorReason** - Indicates the reason for the error that happened.

- 0 – Unknown error
- 1 – Internet not available
- 2 – Universally unique identifier (UUID) validation failed
- 3 - Universally unique identifier (UUID) is null or empty
- 4 – JavaScript injection error where the Office app for Android can't pass authToken to Dime
- 5 – Base WebURL loaded on client is invalid

Office.Android.DocsUI.Views.PremiumFeatureUpsell

This event captures clicks by a free user clicks to view a feature behind the pay wall. The data is used to measure the interaction of users with the contextual upsell experience and understand which features are preferred by the user which drives them to buy a subscription. This helps us invest to improve those preferred set of entry points.

The following fields are collected:

- **featureId** - TCID for premium feature
- **featureName** - Premium Feature Title
- **seePlanButtonClick** - How many times "See plan buttons" gets clicked in upsell UI

Office.Apple.IAPReviewYourSubscriptioniOS

This event captures session-based metadata when the In-App-Purchase (IAP) UI is shown to the user and the buttons the user subsequently interacts with. This data is used to help us understand the friction in the purchase flow and compare it with the funnel of a different purchase experience to understand which experience is better for the user.

The following fields are collected:

- **FlowType** - Integer – Flow from where IAP was launched.
- **Restore** - String – rule tag is logged when restore button is clicked
- **PremiumFeatures** - String – rule tag is logged when "PremiumFeatures" button is clicked
- **Product** - String - The SKU selected by the users

Office.Apple.InAppPurchaseContext

This event measures critical usage telemetry for the point of entry of the in-app purchase screen. The data helps understand and improve the user experience by identifying the preferred entry point for an in-app purchase.

The following fields are collected:

- **context** - String – The flow through which the user landed on the in app purchase page

Office.Apple.Licensing.CommonPaywallControl

This event is used to understand the in-app purchase (IAP) experience for the user. It allows us to ensure IAP performs as expected and helps us understand user issues so we can optimize the IAP experience. Collection occurs through one of the following sub-events.

- **Office.iOS.Paywall.Paywall.Presented** - Data is collected when paywall control is shown to the user. The data is used to build a view to measure the conversion rate at every step and ensure that the user interface is performing as expected with users experiencing minimal friction during the purchase experience.

  The following fields are collected:

  - **entryPoint** - String – The Button/Flow from which Paywall was displayed. Like "Premium Upgrade

Button" or "First Run Flow"
    - **isFRE** - Boolean – Are we showing the First Run Experience or regular UI?
    - **PaywallSessionId** - String – Collected to uniquely identify a Paywall session in an app session
- **Office.iOS.Paywall.Paywall.Stats** - Data is collected when the paywall user interface is shown to the user, the duration of the interaction and whether a purchase was attempted, succeeded, or failed. The data is used to measure the performance of the user interface and ensure that it performing as expected.

  The following fields are collected:

    - **entryPoint** - String – The Button/Flow from which Paywall was displayed. Like "Premium Upgrade Button" or "First Run Flow".
    - **isFRE** - Boolean – Check to see if the First Run Experience or regular UI is showing.
    - **PaywallSessionId** - String – Collected to uniquely identify a Paywall session in an app session
    - **status** - String – Exit status of Paywall. Like "initiated", "paymentDone", "provisionFailed".
    - **userDuration** - Double – Duration in milli-seconds the user spent on Paywall
- **Office.iOS.Paywall.SKUChooser.BuyButtonTap** - Data is collected when user taps the Purchase/Buy Button. The data is used to measure the performance of the button and ensure that it performing as expected.

  The following fields are collected:

    - **entryPoint** - String – The Button/Flow from which Paywall was displayed. Like "Premium Upgrade Button" or "First Run Flow".
    - **isDefaultSKU** - Bool – If the user is purchasing the product, we recommended for them, by displaying it by default.
    - **PaywallSessionId** - String – Collected to uniquely identify a Paywall session in an app session
    - **productId** - String – App-store product-id of the product for which the Buy Button was tapped.
    - **toggleCount** - Int – Number of times the user switched between viewing various products, before they tapped the Buy Button, in the current session of Paywall.
- **Office.iOS.Paywall.SKUChooser.Stats** - Data collected to see how the user entered the SKU Chooser, how much time the user spends on the SKU Chooser screen and why they exited the SKU Chooser. Using the information, we can ensure that the SKU Chooser is performing as expected, and we will be able to optimize and improve the end user experience.

  The following fields are collected:

    - **entryPoint** - String – The Button/Flow from which Paywall was displayed. Like "Premium Upgrade Button" or "First Run Flow".
    - **exitReason** - String – Exit reason of SKU Chooser. Like "BuyButton", "CloseButton"
    - **isFRE** - Boolean – Are we showing the First Run Experience or regular UI?
    - **PaywallSessionId** - String – Collected to uniquely identify a Paywall session in an app session
    - **userDuration** - Double – Duration in milli-seconds the user spent on the SKU chooser.
- **Office.iOS.Paywall.FailedScreen.RetryButtonTap** - Data collected when the Purchase/Provisioning/Activation failed, and the user tapped the retry button. The data is used to troubleshoot purchase error scenarios and fix it to ensure that it performs as expected.

  The following fields are collected:

    - **failureReason** - String – Indicates what the failure was the user is retrying; for example, "provisioningFailed", "purchaseFailed", "activationFailed".
    - **PaywallSessionId** - String – Collected to uniquely identify a Paywall session in an app session
    - **productId** - String – App Store ID of the product for which user is retrying the failed request.
- **Office.iOS.Paywall.SKUChooser.MoreBenefits.Stats** - Data collected when users tap on "See More

Benefits" to see all the services, apps and features included in the purchase. They must expand sections detailing the features for each of the apps. This event collects which features and apps they expanded, along with the duration of time spent. The data is used to ensure that the UI offered to end users to learn about the benefits is performing as expected.

The following fields are collected:

- **appsExpanded** - String – Comma-separated list of services/apps for which the benefits were expanded.
- **PaywallSessionId** - String – Collected to uniquely identify a Paywall session in an app session
- **productId** - String – App Store ID of the product for which user is viewing more benefits offered
- **userDuration** - Double – Duration in milli-seconds the user spent on the Benefits Screen.

- **Office.iOS.Paywall.SuccessScreen.SeeAllBenefitsButtonTap** - This event is collected when the user taps "See All Benefits" after a successful purchase to see the apps and features included in the purchase. The data is used to measure that the user interface is performing as expected.

  The following fields are collected:

  - **PaywallSessionId** - String – Collected to uniquely identify a Paywall session in an app session
  - **productId** - String – App Store ID of the product for which user is viewing all benefits offered.
- **Office.iOS.Paywall.SKUChooser.ProductSwitched** - Usage telemetry to view the end user's interaction with the UI provided to switch between different SKUs and ensure that it is performing as expected.

  The following fields are collected:

  - **PaywallSessionId** - String – Collected to uniquely identify a Paywall session in an app session
  - **productId** - String – App Store ID of the product the user just switched to viewing from the available products on the SKU chooser.
- **Office.iOS.Paywall.StoreKit.Response** - Critical engineering telemetry to log the result of purchase attempt triggered manually by user and the App store response to the event. The data is used to measure the status of a purchase attempt and reasons of failure (if any) and take corrective actions to ensure that the IAP and all the entry points as performing as expected.

  The following fields are collected:

  - **entryPoint** - String – The Button/Flow from which Paywall was displayed. Like "Premium Upgrade Button" or "First Run Flow".
  - **failureReason** - String – Only added when status is "failure". Indicating the error response given by the App-store response.
  - **PaywallSessionId** - String – Collected to uniquely identify a Paywall session in an app session
  - **productId** - String – Only for "MakePurchase", "PendingPurchase", the app-store ID of the product for which the request was made.
  - **productsCount** - Int – Only for "ProductsFetch", the number of products returned by Store.
  - **requestType** - String – Type of StoreKit request. Like "ProductsFetch", "PendingPurchase", "Restore"
  - **status** - String – Success or Failure, indicating success or failure of the request.
- **Office.iOS.Paywall.Provisioning.Response** - Critical Engineering Telemetry and Contract with Retail Federation Service (RFS) to collect the information provided in this. RFS is the internal service used within Microsoft for crosschecking the purchase. This is used to get the health of the API call made to RFS which would help in understand that the performance of the integration is as expected.

  The following fields are collected:

  - **entryPoint** - String – The Button/Flow from which Paywall was displayed. Like "Premium Upgrade

Button" or "First Run Flow".
    - **failureReason** - String – Only added when status is "failure". Indicating the error response given by the RFS Provisioning response.
    - **PaywallSessionId** - String – Collected to uniquely identify a Paywall session in an app session
    - **productId** - String – App Store ID of the product the request was made for
    - **status** - String – Success or Failure, indicating if the request succeeded or failed
- **Office.iOS.Paywall.SignIn.Response** - The event is collected when users complete SignIn during upsell flow, which is triggered for PreSignIn upsell scenarios like the PreSignIn FRE and PreSignInDiamond. This can be used to check the SignIn rates during the Upsell flow and help us analyse the PreSign scenarios.

  The following fields are collected:

    - **entryPoint** - String – The Button/Flow from which Paywall was displayed. Like "Premium Upgrade Button" or "First Run Flow".
    - **PaywallSessionId** - String – Collected to uniquely identify a Paywall session in an app session.
    - **status** - String – The SignIn status of the user. Can be Cancelled, Failure, PremiumSignIn or Success (Non-Premium Signin)

Office.Apple.Licensing.CommonPaywallDetails

This event logs the user details before Paywall control is shown to the user. The data is used to diagnose issues with Common Paywall Control (CPC) and will be used in conjunction with the table Office.Apple.Licensing.CommonPaywallControl to check if there are any issues in the code or to debunk any data anomalies with regard to CPC.

The following fields are collected:

- **canUserSeeUpsell** -Boolean:True if the SignedIn user is not underage and qualified to see upsell screen
- **EffectiveIdentityType** - Boolean: Sign in type of the user. Can be -1 (Unsigned), 1 (MSA), 2 (OrgId)
- **HasSubscription** - Boolean: True if the user has an active Microsoft 65 subscription
- **IsCPCOnSignInEnabled** - Boolean: True if the FeatureGate Microsoft.Office.LicensePurchase.FollowSignInWithCPC is enabled
- **isFREUpsellToUnsignedUsersEnabled** - Boolean: True if the FeatureGate Microsoft.Office.LicensePurchase.FREUpsellToUnsignedUsers is enabled
- **IsProClassDisplay** - Boolean: If the users' device is a pro class display (Screen size > 10.1 inch) or not
- **ShowCPC** - Boolean: If CPC First Run Experience (FRE) is to be shown to the user.
- **SKUEffectiveIdentityType** - Int: Sign in type of the user. Can be -1 (Unsigned), 1 (MSA), 2 (OrgId)
- **SKUHasSubscription** - Boolean: True if the user has an active Microsoft 365 subscription
- **SKUIsCommonPaywallControlEnabled** - Boolean: True if the FeatureGate Microsoft.Office.LicensePurchase.UseCPC is enabled
- **SKUIsPreSignInDiamondEnabled** - Boolean: True if the FeatureGate Microsoft.Office.LicensePurchase.PreSignInDiamond is enabled
- **SKUIsProClassDisplay** - Boolean: If the users' device is a pro-class display (Screen size greater than 10.1 inch)
- **SKUShowCPC** - Boolean: If CPC SKU Chooser is to be shown to the user

Office.Apple.Licensing.PremiumFeatureUpsell

This event is triggered when a free user clicks to view a feature behind the paywall. The data is used to measure the interaction of users with the contextual upsell experience and ensure that it is working as expected.

The following fields are collected:

- **CanUserSeeUpsell** - Captured when the state of the users allow them to see an upsell CTA
- **dismissUpsellUI** - Captured when users click on "Cancel Button" in alert box or user dismiss the bottom sheet to
- **featureId** - Identifier for the premium feature which users are trying to use
- **learnMoreButtonClick** - Captured when users click on "Learn More" button
- **LicensingUpgradeUIShown** - Captured when users see the upsell alert box
- **seePlanButtonClick** - Captured when users click on "See plans" button

Office.Dime.Sdk.Health

This event captures data that helps in monitoring the health of the Dime components. For example, for the in-app purchase flow when a user opts to buy a Microsoft 365 subscription from within the Office app for Android or on devices running Windows.

The following fields are collected:

- **Data_ActivityErrorDescription** - Error description of the activity
- **Data_ActivityErrorMessage** - Error message of the activity
- **Data_CampaignId** - Campaign ID for attribution
- **Data_ContentId** - Based on the Experience ID; it is mapped to a Flow ID and a Content ID
- **Data_CorrelationVector** - Correlation Vector to correlate dime with partners that use correlation vector
- **Data_CustomerImpacted** - Used for troubleshooting if customer is impacted in loading the flow
- **Data_DimeActivityDuration** - Duration time
- **Data_DimeActivityMetadata** - Activity metadata
- **Data_DimeActivityName** - Activity name for health monitoring
- **Data_DimeActivityResult** - Activity result, Success/ Error/ Expected Error
- **Data_DimeVersion** - Build Version
- **Data_DurationLevel** - Severity - 0/1/2
- **Data_EcsConfigIds** - IDs for the experiments
- **Data_EcsCountry** - Detected country
- **Data_EcsETag** - Flights information
- **Data_Environment** - Dime Environment production/pre-production
- **Data_ExperienceId** - Experience to load
- **Data_FlowId** - Based on the Experience ID; it is mapped to a Flow ID and a Content ID
- **Data_Language** - Culture
- **Data_Market** - Detected market

- **Data_OTelJS_Version** - Office telemetry version
- **Data_PageSessionId** - Session ID of the page
- **Data_PartnerId** - Caller App
- **Data_QosLevel** - Severity 0/1/2
- **Data_SDX_AssetId** - Asset ID of the Service Delivered Experience (SDX) hosting content for Win32
- **Data_SDX_BrowserToken** - Token of the browser for Win32
- **Data_SDX_HostJsVersion** - JavaScript library version for Win32
- **Data_SDX_Id** - Service Delivered Experience Id for Win32
- **Data_SDX_InstanceId** - Instance ID of the SDX for Win32
- **Data_SDX_MarketplaceType** - SDX Marketplace Type for Win32
- **Data_SDX_OfficeJsVersion** - Office JS Version for Win32
- **Data_SDX_SessionId** - Session ID of the SDX for Win32
- **Data_SDX_Version** - SDX Version for Win32
- **Data_TimestampUTC** - Timestamp of the event
- **Data_TsgId** - Troubleshooting Guide Id for each activity
- **Data_UserAgent** - Header Tags

Office.Docs.Shared.PremiumFeatureMessageBar

This event collects free users' taps on a premium feature that resides behind the paywall. The data is used to understand the set of features consumers are interacting with as they convert to a paid user. This tells us the preferred entry points of the users and improve the user experience.

The following fields are collected:

- **featureId** - TCID for premium feature on which user taps

Office.Licensing.AcceptEulaForCurrentLicense

This is collected when the user gets licensed and accepts EULA for the current license

It is used to detect if the user is in a good state or not, used for system health and used for diagnostic purposes if a user reports an issue with their machine

The following fields are collected:

- **ACID** - A GUID identifier representing the Office product that the user is licensed for
- **DwEulaId** – Numeric identifier of the type of EULA that was accepted by the user

Office.Licensing.ActivateDeviceEntitlement

This event gets triggered when we are trying to activate a device-based perpetual Office offer for the user. We use this data to monitor the health of the systems and services.

The following fields are collected:

- **Activity_Success** - tells us if the device is licensed with a device-based perpetual Office offer.
- **Data_Count** - tells us the number of device-based perpetual Office entitlement associated with the device. Technically, there should be no more than one.

- **Data_EligibleEntitlementsCount** - tells us the number of eligible entitlements. Because service will return all the device entitlements associated with the device, but we need to check those offers that are relevant to the Office application that's being running.
- **Data_Errors** - a string with a list of errors while we're fetching licenses for the entitlements, separated by comma.
- **Data_LicensedEntitlementsCount** - tells us the number of entitlements that we successfully fetch a license for. There could be entitlement errors that lead us to not be able to get a license.

Office.Licensing.Activation

Post setting up the license on the machine, we attempt to activate the license by calling the AVS service. This reports the result of the activation call

It is critical in detecting how many users are facing activation issues. We have anomaly detection to detect any regression. This is super critical as we have an external dependency on AVS and this signal points whether our external partners are healthy. It is also used for diagnostic purposes and system health if a user reports an issue with their machine

The following fields are collected:

- **Acid** - A GUID identifier representing the Office product that the user is licensed for
- **ReferralData** – Identifier of the OEM that installed Office on the machine

Office.Licensing.ActivationWizard

If we are not able to automatically activate the license for some reason, we show an activation wizard to the user. This reports that the wizard is being shown to the user. It is critical in detecting if the user is in a good state and not missing functionality, used for system health and used for diagnostic purposes if a user reports an issue with their machine

This event collects no fields.

Office.Licensing.BusBar.CheckForDynamicBusbarExperiment

This event is triggered once for every licensing business bar type that will be shown that has the dynamic business bar flight on (treatment group). This data event reports whether there is a Lifecycle Programming Platform dynamic business bar campaign ready on disk. Data will be used to measure the health of the new Lifecycle Programming Platform dynamic licensing business bar technology.

The following fields are collected:

- **DoesCampaignExist (bool)** - Indicates if the campaign is on disk
- **Type (int32)** - Indicates the licensing business bar type

Office.Licensing.BusBar.ShowStashedBusbar

This event is triggered when the dynamic Lifecycle Programming Platform business bar fails to show and stashed static business bar needs to be shown instead. This data event will be used to make sure fallback to static business bar is successful.

The following fields are collected:

- **Type (int32)** - Indicates the licensing business bar type

Office.Licensing.Dialogs.WebViewDialog.Close

This event is used as a signal to tell us that the in-app purchase experience is being closed either by the user or by the application. The data is used to monitor and alert on the health of the in-app purchase flow to ensure it is working as expected.

The following fields are collected:

- Data_ClosedDialog - flag indicating the user closed the dialog

Office.Licensing.Dialogs.WebViewDialog.HandleErrorNotification

This event is used as a signal to tell us that the in-app purchase experience attempted to load, but an error occurred which resulted in the dialog not showing. The data is used to monitor and alert on the health of the in-app purchase flow to ensure it is working as expected.

The following fields are collected:

- Data_MoeErrorCode - Error code seen in the web dialog framework

Office.Licensing.Dialogs.WebViewDialog.Preload

This event is used as a signal to tell us that the in-app purchase experience is being loaded in the background. The data is used to monitor and alert on the health of the in-app purchase flow to ensure it is working as expected.

The following fields are collected:

- None

Office.Licensing.Dialogs.WebViewDialog.Show

This event is used as a signal to tell us that the in-app purchase experience is being shown to the user. The data is used to monitor and alert on the health of the in-app purchase flow.

The following fields are collected:

- None

Office.Licensing.Dialogs.WebViewDialog.Timeout

This event is used as a signal to tell us that the in-app purchase experience attempted to load but timed out. The data is used to monitor and alert on the health of the in-app purchase flow to ensure it is performing as expected.

The following fields are collected:

- None

Office.Licensing.EnforceSignInQualified

This is the signal that tells us if the experiment that we are running to enforce user sign as part of licensing is successful. This is critical in detecting the success or failure of the experiment that is forcing the users to log in which is a required step for the modern licensing stack. Failure to sign in will result in the users not being able to use the app.

The following fields are collected:

- Qualified – Identifies whether the user qualified for the sign in enforcement

Office.Licensing.ExpirationDialogShown

This is collected when we show the expiration dialog to the user that says that their license has expired. It is critical in detecting if the user is in a good state and not missing functionality, used for system health and used for diagnostic purposes if a user reports an issue with their machine

The following fields are collected:

- LicNotificationState – An enumerator that tells us what kind of notification is being shown to the user

Office.Licensing.FullValidation

This is collected on every session that reports the licensing state of the machine and reports the errors that the user is seeing due to which they are not able to use the application. This event indicates if the user's machine is healthy or not. We have anomaly detection set up for this event to indicate if a regression or activation mechanism is causing bad user behavior. This is also critical when diagnosing user issues and for monitoring system health.

The following fields are collected:

- **Acid** – A GUID identifier representing the Office product that the user is licensed for
- **ActivationAttributes** - Type of activation mechanism that the user is using.
- **IsSessionLicensing** – Whether we are currently running under shared computer activation mode or not
- **LicenseCategory** – Category of the Office license that the user is using
- **Licenses** – List of names of all the Office licenses that are present on the machine
- **LicenseStatuses** – Status of all the Office licenses present on the machine

Office.Licensing.GetEntitlement

We collect this when the user is setting up a device and we call our licensing service to detect if the logged in user has an Office entitlement or not. This reports the result of that call. It is critical in detecting if the user is in a good state and missing functionality, used for system health and used for diagnostic purposes if a user reports an issue with their machine

The following fields are collected:

- **EntitlementCount** – Number of entitlements the user has

Office.Licensing.GetNextUserLicense

This event gets triggered when fetching license for the chosen user-based Office entitlement during the activation first run experience. We use this data to monitor the health of the systems and services.

The following fields are collected:

- **Activity_Success** - Boolean: tells us if we successfully fetched a license for the device to be activated on its Office application.
- **Data_AllowNULPerpetual** - Boolean: tells us if the flight to enable vNext Perpetual is on.
- **Data_AttemptNulReactivation** - Boolean: tells us if this is a reactivation scenario.
- **Data_CurrentMode** - 0 means SPP (the legacy licensing stack), 2 means vNext (the modern licensing stack).
- **Data_HasError** - Boolean: tells us if we got any error when trying to get a license for the chosen user-based entitlement.
- **Data_IsSubscription** - Boolean: tells us if the reactivation is for subscription office.
- **Data_NewMode** - 0 means SPP (the legacy licensing stack), 2 means vNext (the modern licensing stack). We should expect 2 most of the time.
- **Data_SkuToSkuNeeded** - Boolean: tells us if we need to do SKU to SKU conversion due to the entitled Office SKU not matching the installed Office SKU.

Office.Licensing.Heartbeat

On every session, we check if 72 hours have passed since the last license renewal and try to extend the expiry of the current license. This event reports the success or failure of the call that we make to ensure we can extend the expiry of the license and keep the user's Office installation functional. It is critical in diagnosing subscription-

related issues and service issues for the user and is critical in detecting regressions for already activated subscription users.

The following fields are collected:

- **Mode** – An enumerator representation of the Office licensing stack that is being used on this machine

Office.Licensing.InClientPinRedemption.CallPinRedemptionAPI

This telemetry tracks the results of Office's pin redemption service call.

The following fields are collected:

- **ClientTransactionId** - Unique identifier for the service call.
- **ErrorCategory** - Each error type can fall into a more general category, such as "Retryable".
- **ErrorType** - Reason of failure, such as "AlreadyRedeemedByOther".
- **InAFOFlow** - A Boolean indicating if we are in the Activation for Office redemption flow.
- **StatusCode** - One-word result of the service call, such as "Created".
- **StatusMessage** - Details of the status code, such as 'Successfully provisioned."
- **UsingNulApi** - A Boolean indicating if we are using the new licensing stack.

Office.Licensing.InRFM

If the device enters reduced functionality mode, we send out this signal to indicate that the machine is not in a healthy state. It is critical in detecting if the user is in a good state and missing functionality, used for system health and used for diagnostic purposes if a user reports an issue with their machine

The following fields are collected:

- **ACID** - A GUID identifier representing the Office product that the user is licensed for
- **DaysRemaining** - Number of days remaining before the current Office license expires
- **Mode** – An enumerator representation of the Office licensing stack that is being used on this machine
- **ProductName** – Name of the product that the user is currently using
- **Reason** – The error code indicating the reason for the current status of the license

Office.Licensing.InstallKey

This is collected when we try to install a key on the device to license the machine. It reports whether the installation was successful and if it was not then the error code. It is critical in detecting if the user is in a good state and not missing functionality, used for system health and used for diagnostic purposes if a user reports an issue with their machine

The following fields are collected:

- **Prid** – Name of the product group for which a key is being installed
- **SkuId** - A GUID identifier representing the Office product for which a key is being installed

Office.Licensing.InvokeLicenseWizard

In case we see problems with the Activation workflow, we trigger a license wizard and send out this signal to indicate the same. It is critical in detecting if the user is in a good state and not missing functionality, used for system health and used for diagnostic purposes if a user reports an issue with their machine

The following fields are collected:

- **Acid** - A GUID identifier representing the Office product that the user is licensed for
- **LicenseStatus** – Status of the Office license that the user is using
- **MachineKey** - An alphanumeric identifier of the license key that was issued to the user

Office.Licensing.LaunchSetupOffice

This event is triggered when we redeem an Office offer for the user who either bought a device bundled with an OEM Office pre-entitlement or has entered a product key. We use this data to monitor the health of the systems and services.

The following fields are collected:

- **Activity_Result_Tag** - tells us how we finished this event.
- **Data_DialogResult** - tells us the overall result of the redemption process.
- **Data_Scenario** - tells us the scenario where the redemption occurred for.

Office.Licensing.LicensingBar

If the device is facing licensing issues and we end up showing a busbar to the user, we send out this signal which also reports the type of busbar shown to the user. It is critical in detecting if the user is in a good state and not missing functionality, used for system health and used for diagnostic purposes if a user reports an issue with their machine.

The following fields are collected:

- **SuppressNotification** – Indicates if we suppressed the licensing busbar
- **Title** – The title of the licensing busbar that was shown to the user
- **Type** – The type of the licensing busbar shown to the user

Office.Licensing.LicExitOfficeProcess

If weend up closing or crashing Office due to a licensing issue, we send out this signal to indicate the same. It is critical in detecting if the user is in a good state and not missing functionality, used for system health and used for diagnostic purposes if a user reports an issue with their machine.

The following fields are collected:

- **ExitCode** – The internal code which caused the app to exit

Office.Licensing.LoadIdentityTicket

In the process of trying to license the device, the app tries load the user's identity in order to see if the user has Office entitlement or not. This event reports the success or failure of the same along with the error code. It is critical in detecting if the user is in a good state and not missing functionality, used for system health and used for diagnostic purposes if a user reports an issue with their machine.

The following fields are collected:

- **FederationProvider** – A string that identifies the federation provider for the currently logged in user
- **IdentityProvider** – A string that identifies the identity provider for the currently logged in user

Office.Licensing.LVUX.EULAExplicitCrash

This is collected if we showed the EULA to the user and the user chose to not accept it as a result of which we crash/close the app. It is critical in detecting if the user is in a good state and not missing functionality, used for system health and used for diagnostic purposes if a user reports an issue with their machine.

The following fields are collected:

- **Acid** - A GUID identifier representing the Office product that the user is licensed for
- **OptInShown** – Indicates whether the opt-in dialog shown on the first boot of the app has already been shown

Office.Licensing.NextUserLicensingEligible

This signal tells us if a user is qualified to move to our new licensing stack. This is critical to quantify the impact on existing users as we roll out our new licensing stack and make sure that users are not losing functionality.

This event collects no fields.

Office.Licensing.Nul.Fetcher.FetchModelFromOls

When the device is on the modern licensing stack, we try to get a license file directly from the service. This event reports the success or failure along with the error code of that service call. It is critical to detect if the user is in a good state on the modern licensing stack, used for system health and used for diagnostic purposes if a user reports an issue with their machine.

The following fields are collected:

- **MetadataValidationResult** – Result of the validation of the metadata of the license to verify that it has not been tampered with
- **SignatureValidationResult** – Result of the validation of the signature of the license to verify that it has not been tampered with

Office.Licensing.Nul.Validation.FullValidation

This is collected on every session of a device that is running on the modern licensing stack. It reports the licensing state of the machine and reports the errors that the user is seeing due to which they are not able to use the app. This event indicates if the user's machine is healthy on the modern licensing stack. We have anomaly detection set up for this event to indicate if a regression is causing bad user behavior. This is also critical when diagnosing user issues and for monitoring system health.

The following fields are collected:

- **Acid** - A GUID identifier representing the Office product that the user is licensed for
- **AllAcids** – List of all the GUIDs of the product that the user is currently licensed for
- **Category** - Category of the Office license that the user is using
- **DaysRemaining** – Number of days remaining before the current Office license expires
- **LicenseId** – An alphanumeric identifier of the license that was issued to the user
- **LicenseType** - Type of the Office license that the user is using

Office.Licensing.OfficeClientLicensing.DoLicenseValidation

This is licensing metadata that is collected from the device on every boot that reports the license acid, license status, type and other properties of the license that are critical in identifying the features set available to the user. This is critical in identifying the feature set available to the user and if any functionality is missing for the user. It is also used for Daily active users/Monthly active user calculations and various other reports by various teams across Office as this tells the type product that the user is using, whether it is a subscription product and whether they are missing any critical functionality.

The following fields are collected:

- **FullValidationMode** – Mode indicating that we are in full validation of license verification
- **IsRFM** – Indicates whether the user is in reduced functionality mode or not

- **IsSCA** – Indicates whether we are running in Shared computer activation mode
- **IsSubscription** – Indicates whether the user is using a subscription license or not
- **IsvNext** – Indicates whether we are using the new modern licensing stack or not
- **LicenseCategory** - Category of the Office license that the user is using
- **LicenseStatus** – Status of the Office license that the user is using
- **LicenseType** - Type of the Office license that the user is using
- **LicensingACID** - A GUID identifier representing the Office product that the user is licensed for
- **OlsLicenseId** - An alphanumeric identifier of the license that was issued to the user
- **SkuIdIsNull** – Indicates whether we encountered an error and don't know the product that the user is running
- **SlapiIsNull** – Indicates whether we encountered a problem in populating one of the licensing objects

Office.Licensing.OnlineRepair

If we are not able to activate a user for some reason and have to show them a dialog that asks them to go online and try repair steps, this event is fired. It is critical in detecting if the user is in a good state and not missing functionality, used for system health and used for diagnostic purposes if a user reports an issue with their machine

This event collects no fields.

Office.Licensing.OOBE.HandleDigitalAttachFailure

This event is triggered when the service check (see Office.Licensing.OOBE.SearchForDigitalAttach) didn't find a claimable Digital Attach offer on this device. Based on the different conditions of the device, we will show different dialogs to user. This event will log various scenarios on how we handle digital attach failure.

The following fields are collected:

- **Activity_Result_Tag** tells us how we transition user to various error states.
  - 0x222e318f - We should keep searching Activation for Office offer.
  - 0x222e318e - We will fall back to OEM mode in this session for when this device doesn't come with any Digital Attach offer.
  - 0x222e318d - No internet connectivity, which will lead us to show NoInternetConnectivity dialog to user
  - 0 - We will show various error UI to user based on their specific error code.
- **Data_DigitalAttachErrorType** - tells us what the specific error code is from the service call.
- **Data_FallbackFlight** - tells us if the UseAFOAsFallBack flight is turned ON or not.

Office.Licensing.OOBE.HandleDigitalAttachSuccess

This event is triggered when the service check finds a claimable Digital Attach offer on this device. Based on the different conditions of the device, we will show different dialogs to user. This event will log various scenarios on how we handle digital attach success.

The following fields are collected:

- **Activity_Result_Tag** - tells us how we handle the Digital Attach Success scenarios.
  - 0 - We're able to auto load identity and we've shown the "You've got Office" (with account) UI to the user.

- 0x222e3191 - We're not able to auto load identity, so we're going to show them the "You've got Office" (without account) UI.
- 0x222e3193 - We've shown the "You've got Office" (without account) UI to the user, or we don't need to show any "You've got Office" UI to the user because it's a device-based offer.

- **Data_IsClaimTypeDevice** - tells us if the claim type of the digital attach offer is device based.

Office.Licensing.OOBE.PopulateDigitalAttachOfferSignInDEX

Original Equipment Manufacturers (OEM) sell machines that come with Office (one-year subscriptions or perpetual) which are paid for when the customer purchases their machine. This event tracks when an Office pre-entitlement is found for the device and the user is already signed in with a Microsoft Account to allow us to monitor the health of the system and services.

The following fields are collected:

- **Data_ExpirationDate** - tells us the expiration date for the subscription offer
- **Data_IsSubscription** - tells us if the to-be-claimed product is subscription SKU or perpetual SKU
- **Data_ProductName** - tells us the product name of the digital attach offer

Office.Licensing.OOBE.SearchForDigitalAttach

Original Equipment Manufacturers (OEM) sell machines that come with Office (one-year subscriptions or perpetual) which are paid for when the customer purchases their machine. Machines that are setup with a specific regkey (OOBEMode: OEMTA) might have an Office offer digitally attached to it. When we boot Office, we perform service checks to see if a digitally attached Office offer is found. This activity is instrumented to track this.

The following fields are collected:

- **Activity_Result_Tag** - tells us the overall result of this service check.
  - 0x222e318c Digital Attach Flight is turned OFF, so no service check is made.
  - 0x222e318b The client doesn't have internet, so no service check is made.
  - 0x222e318a Found a redeemable Digital Attach offer
  - 0x222e3189 Found a non-redeemable Digital Attach offer
- **Data_EnableDAFlight** - tells us if the Digital Attach flight that enables this service check is ON or not.

Office.Licensing.OOBE.ShowTouchlessAttachFailureDialog

Original Equipment Manufacturers (OEM) sell machines that come with Office (one-year subscriptions or perpetual) which are paid for when the customer purchases their machine. This event is triggered when an error occurs in the Digital Attach redemption and activation flow for OEM PCs that come pre-entitled with Office. We use this data to monitor the health of the systems and services and fix issues related the OEM Office activation flow.

The following fields are collected:

- **Data_Continue** - tells us if user clicks "Continue" on the dialog.
- **Activity_Result_Tag** - tells us the button the user clicked on the dialog.
  - 0x222e319d - User clicks "Retry" on the dialog
  - 0x222e319c - User clicks "Continue" on the dialog
  - 0 - User exits out of the dialog
- **Data_IsForDigitalAttach** - tells us which platform and workflow the user is on – Legacy (Activation for Office (AFO)) or Modern (Digital Attach).
- **Data_Retry** - tells us if the user clicks "Retry" on the dialog.

Office.Licensing.OOBE.ShowTouchlessAttachOfferDialog

Original Equipment Manufacturers (OEM) sell machines that come with Office (one-year subscriptions or perpetual) which are paid for when the customer purchases their machine. This event tracks when an Office pre-entitlement is found for the device and the user is not signed in with a Microsoft Account to allow us to monitor the health of the system and services.

The following fields are collected:

- **Activity_Result_Tag** - tells us if an identity was found for the user
  - 0x222e3194 - We're not able to get user identity (they either cancelled sign-in or authentication failed).
  - 0 - We got an identity from user.
- **Data_ExpirationDate** - tells us the expiration date for the subscription offer
- **Data_IsCentennial** - tells us if the running office application is on centennial platform
- **Data_IsForDigitalAttach** - tells us if this dialog is triggered from Digital Attach flow or Activation for Office flow.
- **Data_IsSubscription** - tells us if the to-be-claimed product is subscription SKU or perpetual SKU
- **Data_OExType** - tells us if user exits out the dialog after they click ChangeAccount link
- **Data_ProductName** - tells us the product name of the digital attach offer
- **Data_UseInAppRedemption** - tells us if we use in-app redemption of web redemption – this is only relevant to Activation for Office flow.

Office.Licensing.OOBE.TryBuyChoice

Users with pre-installed Office on new machines who have no Office entitlement are shown a dialog through which they can try, buy or enter a product key to get licensed. This event captures the user action on the dialog. This event is used to track the user action taken on the dialog shown to users with no Office entitlement where Office was pre-installed on the machine and helps determining if the user is licensed or unlicensed by design.

The following fields are collected:

- **Buy** - Tells if the user clicked on the buy button or not
- **ForceAutoActivate** - Tells if in-app activation should be attempted or not
- **GoBackToSignIn** - Tells if the user wanted to sign in again (possibly with another account)
- **IsPin** - Tells if the user entered a pin
- **ProductKey** - Tells if the user tried to enter a product key
- **Try** - Tells if the user clicked on the try button or not
- **UserDismissed** - This tells if the user dismissed the dialog and thus would be in grace or reduced functionality mode because they didn't choose to buy office or get a trial

Office.Licensing.Purchase

[This event has been removed from current builds of Office, but might still appear in older builds.]

We have an experiment that gives the user an option to try and set up autopay for Office directly from an app without ever leaving the context of the app. This reports the success or failure of that experiment along with the error code It is critical in detecting if the user is in a good state and not missing functionality, used for system health and used for diagnostic purposes if a user reports an issue with their machine.

The following fields are collected:

- **StorePurchaseStatus** – Represents the error code/success code of the purchase call that was made through windows store

Office.Licensing.SearchForSessionToken

If the user is running under shared computer activation mode, we try to search for a session token on the machine which allows the user to use the app. This event reports the success or failure of the scenario along with the error code. It is critical in detecting if the user is in a good state and not missing functionality, used for system health and used for diagnostic purposes if a user reports an issue with their machine.

The following fields are collected:

- **LoadLicenseResult** – Represents the error code/success code of whether we were able to load the licenses for the current user
- **OpportunisticTokenRenewalAttempted** – Indicates whether we attempted to renew the user's session token opportunistically
- **SetAcidResult** – Represent the error code/success code of whether we were able to set the acid to the expected value

Office.Licensing.ShowNewDeviceActivationDialog

On the first boot of an Office app, we will try to show a sign in dialog pre-populated with the credentials the user employed to download Office. The user can then continue to sign-in with those credentials, use different credentials or dismiss the dialog. This event reports the action taken by the user when presented with this dialog. It is critical for detecting if a user is in a good state on the modern licensing stack, used for system health and used for diagnostic purposes if a user reports an issue with their machine

The following fields are collected:

- **UserAction** – Identifier for the action taken by the user when presented with the dialog.

Office.Licensing.SkuToSkuConversion

As part of licensing the user, if we have to change the user's SKU from one SKU to another, we send this signal out along with the success or failure code. It is critical in detecting if the user is in a good state and missing functionality, used for system health and used for diagnostic purposes if a user reports an issue with their machine.

The following fields are collected:

- **DestinationSku** – Name of the SKU to which the currently installed product should be converted to
- **PendingAcid** – ID of the product for which a SKU conversion is pending
- **SourceSku** – Name of the original SKU that was installed on the machine
- **UninstallProduct** – Indicates whether the old product will be uninstalled as part of the conversion

Office.Licensing.TelemetryFlow.OLSResults

When a user is unlicensed, we make several service calls to get the user into a licensed state and to activate their Office product. This event gets triggered on calling the Office Licensing Service to check if the user has entitlements. This event is going to be used to track the user licensing health after calling the Office Licensing Service and the Office Client health after attempting to get Office activated.

The following fields are collected:

- **EntitlementPickerShown** - Tells if the user had multiple entitlements and if the user had to manually choose from them to get licensed

- **GetAuthResult** - Tells various states the client might be in like if they got an empty product key from the Office Licensing Service or if they were entitled for another product and Office needs to be converted to the new product
- **GetEntitlementsCount** - Tells the number of entitlements the user has
- **GetEntitlementsSucceeded** - Tells if the call to an Office Licensing Service API to retrieve the user's entitlements succeeded or not
- **GetKeySucceeded** - Tells if the call to an Office Licensing Service API to retrieve a key succeeded
- **GetNextUserLicenseResult** - Tells if the modern licensing stack was able to work and if the user got licensed or not
- **InstallKeyResult** - Tells various reasons why the user might be in a bad state like if activation failed or the installation of the key failed
- **NotInitializedBeforeWhileAdding** - This is just informational and tells if the event was added to a telemetry manager map without explicitly registering for it
- **NotInitializedBeforeWhileSending** - This is just informational and tells if the event was attempted to be sent without explicitly registering for it in the telemetry manager map before hand
- **SentOnDestruction** - This is just informational and tells if the event was added to a telemetry manager map and wasn't sent explicitly
- **Tag** - Used for telling where in the code the event was sent from
- **VerifyEntitlementsResult** - Tells various states the user might be in after validating the entitlements retrieved from the Office Licensing Service

Office.Licensing.TelemetryFlow.SearchForBindingResult

OEMs sell machines that come with Office (one-year subscriptions or perpetual). These Office products are paid for when the customer purchases their machine. Machines that are set up with a specific regkey (OOBEMode: OEMTA) might have an Office binding associated with it. When we boot Office on such machines, we perform service checks to see if an Office binding corresponding to the machine is found.

This telemetry activity tracks the success and failure points in searching for a binding so that we can ensure that machines that do have a binding can successfully fetch them, and that our services are healthy. This activity does not track machines that turn out to not have any bindings associated with them after we check with our services.

The following fields are collected:

- **DexShouldRetry** - Signal that we've hit a retryable issue (no internet or servers are down)
- **GenuineTicketFailure** - Tells us the failure HRESULT when trying to get the machine's Windows genuine ticket/product key (WPK).
- **PinValidationFailure** - Tells us why the pin validation process failed. Possible errors:
  - GeoBlocked
  - InvalidFormat
  - InvalidPin
  - InvalidState
  - InvalidVersion
  - Unknown
  - Used
- **PinValidationResult** - Tells us the pin validation result of a pin that we failed to crack.

- **Pkpn** - The pkpn range that the pin belongs to.
- **Success** - Indicates that we successfully fetched a valid Office binding (pin) for the machine.
- **Tag** - Tells us at which step we stopped searching for a binding. Possible tags:
  - 0x03113809 No internet/service error while validating pin
  - 0x0311380a Pin validation failure, sent with PinValidationFailure field
  - 0x0310410f Success, sent with Success field
  - 0x0311380d Retry-able errors (internet issues, unknown errors)
  - 0x0311380e Non-retry-able errors (binding offer expired)
  - 0x0311380f Other errors (unable to license)
  - 0x03104111 Failed to crack the Office pin, sent with PinValidationResult field
- **WpkBindingFailure** - Tells us the error code of getting the Office pin bound to the machine's WPK.

Office.Licensing.TelemetryFlow.ShowAFODialogs

After successfully obtaining a valid Office pin bound to a machine pre-bundled with Office, we show the user either a sign-in dialog or a redemption dialog. Once the pin is redeemed, we show the EULA dialog. As a part of our modernizing Activation for Office feature, we refreshed the two dialogs to convey more information regarding the Office product that comes with the machine. This telemetry is to track if our feature successfully reduces user friction in redeeming their product by tracking the flow and exit points of the redemption process (which dialog was dismissed).

The following fields are collected:

- **ActionActivate** - Signal that user clicked the “Activate” button.
- **ActionChangeAccount** - Signal that user clicked the “Use a different account” hyperlink.
- **ActionCreateAccount** - Signal that user clicked the “Create account” button.
- **ActionSignIn** - Signal that user clicked the “Sign in” button.
- **CurrentView** - The type of dialog the user closed.
- **DialogEULA** - Signal that we showed the ‘Accept EULA’ dialog.
- **DialogRedemption** - Signal that we showed the Activation for Office redemption dialog.
- **DialogSignIn** - Signal that we showed the Activation for Office sign-in dialog.
- **EmptyRedemptionDefaults** - Signal that we failed to fetch default redemption information.
- **GetRedemptionInfo** - Signal that we’re fetching demographic information for pin redemption.
- **MalformedCountryCode** - Signal that the country code needed for pin redemption is malformed.
- **OExDetails** - The error details we get back when identity's sign-in dialog was dismissed.
- **OExType** - The error type we get back when identity's sign-in dialog was dismissed.
- **Tag** - Tells us at which step the user exits the Activation for Office redemption process. Possible tags:
  - 0x0311380b User dismissed identity's sign-in dialog from redemption dialog
  - 0x0311380c Failed to auto-load an identity post user sign-in from redemption dialog
  - 0x03113810 Failed to load the account's demographic information (country code, language, currency, trial offer, and marketing preferences)
  - 0x03113805 User dismissed identity's sign-in dialog from sign-in dialog
  - 0x03113806 Failed to auto-load an identity post user sign-in from sign-in dialog

    - 0x03113807 Failed to auto-load an identity
    - 0x03113811 User closed the sign-in/redemption dialog
    - 0x03113812 User closed the accept EULA dialog
    - 0x03113808 User accepted the EULA
    - 0x03113811 User closed a dialog
    - 0x2370e3a0 User closed a dialog
    - 0x2370e3c1 Go to web for pin redemption
    - 0x2370e3a1 Go to web for pin redemption
    - 0x2370e3c0 Dialog sequence looped caused by user going back and forth in the dialog flow
    - 0x2370e3a3 User clicked "Not now" hyperlink which skips the Activation for Office offer for that session
    - 0x2370e3a2 User clicked on "Never show this to me" hyperlink which disables the Activation for Office offer
- **UseInAppRedemption** - Tells us if we're keeping users in-app for redemption or sending them to the web to redeem their fetched pin (pre-populated).
- **UseModernAFO** - Tells us if we're using the new or old Activation for Office experience.

Office.Licensing.TelemetryFlow.ShowTryBuyDialogForOOBE

When new machines have Office pre-installed and the user doesn't have an entitlement we show a dialog which gives the user the option to try, buy or enter a product key so that the user can get licensed and this event tracks if the dialog was shown. This event will help with knowing if the dialog was shown to the user to try, buy or enter the product key and hence will help us determine if the user had the opportunity to get licensed.

The following fields are collected:

- **ActiveView** - Tells the dialog ID shown to the user
- **CurrentOOBEMode** - Tells the pre-install mode (OOBE mode such as Activation for Office, OEM etc.)
- **NotInitializedBeforeWhileAdding** - This is just informational and tells if the event was added to a telemetry manager map without explicitly registering for it
- **SentOnDestruction** - This is just informational and tells if the event was added to a telemetry manager map and wasn't sent explicitly
- **ShowTryButton** - Tells if the Try Button was shown to the user on the dialog or not
- **Tag** - Used for telling where in the code the event was sent from

Office.Licensing.TelemetryFlow.TrialFlow

When an unlicensed user of Office pre-installed on a machine is attempting to get a trial, this event gets triggered. It is used to see which path the user would follow to get a trial and if there were any errors while getting the trial through in-app purchases. Depending on the user action and the result of the in-app purchase the user could end up being unlicensed.

The following fields are collected:

- **HasConnectivity** - Tells if the user has internet connectivity and in case there isn't the user might have to use a grace license for five days or may be in reduced functionality mode
- **InAppTrialPurchase** - Tells if the flight is enabled for launching the Store Purchase SDK to capture PI and purchase a trial from within the application *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **IsRS1OrGreater** - Tells if the OS Version is greater than RS1 or not since the Store Purchase SDK should

be used only if the OS Version is greater that RS1

- **NotInitializedBeforeWhileAdding**: This is just informational and tells if the event was added to a telemetry manager map without explicitly registering for it
- **OEMSendToWebForTrial** - Tells if the flight is enabled to send users to the web to redeem a trial
- **StoreErrorConditions** - Tells the various conditions under which the Store Purchase SDK could have failed *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **StoreErrorHResult** - Tells the error code returned from the Store Purchase SDK *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **StorePurchaseStatusResult** - Tells the result of calling the Store Purchase SDK and if the user made a purchase or not which will help determine if the user should get licensed to use Office *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **Tag** - Used for telling where in the code the event was sent from
- **UserSignedInExplicitly** - Tells if the user signed in explicitly in which case, we would re-direct users to the web for the trial *[This field has been removed from current builds of Office, but might still appear in older builds.]*

Office.Licensing.UseGraceKey

For some reason if we are unable to license the user, we install a grace key and send out this signal that signifies. It is critical in detecting if the user is in a good state and missing functionality, used for system health and used for diagnostic purposes if a user reports an issue with their machine

The following fields are collected:

- **OpportunisticTokenRenewalAttempted** – Indicates if we attempted an opportunistic renewal for the user in shared computer activation mode
- **ReArmResult** – Indicates the result of rearming the installed key which can extend the expiry of the current license

Office.SetupOffice.Sdx.Log

This event is triggered when we redeem an Office offer for the user who either bought a device bundled with an OEM Office pre-entitlement or has entered a product key. This data is used for general log messages.

The following fields are collected:

- **Ctid (Data_Ctid)** - CorrelationId is used for linking different logs withing one redemption session.
- **Environment (Data_ Environment)** - dev environment (Pr, Edog, Prod, Int).
- **Message (Data_Message)** - The log message from setup.office.com. For example, "image '../img/spinner.csv' can't be loaded, cdn is used."
- **Type (Data_Type)** - The type of log message (Error, Warning, Info)

OneNote.EnrollmentResult

This event logs the status upon Intune enrollment. This scenario is specific to Intune enabled accounts.

The following fields are collected:

- **EnrollmentResult** - The result of Intune enrollment

SKU.PRODUCT.PRICE.NULL.EVENT

This event is used to capture events to quantify the impact of the bug due to which users today see "Null"

instead of a price at the SKU chooser screen. The bug will be diagnosed further to determine a fix.

The following fields are collected:

- **PriceNotFound** - Prices is not found from the store.
- **StoreNotInitilized** - When store is not initialized successfully.

Microsoft AutoUpdate (MAU) events

additionalappinfo.invalidpreference

This event reports on invalid preference set to display more information with respect to End of Service for a product. We use this information to advise customers to set correct preferences in order to see additional information.

The following fields are collected:

- **App** – The application process sending the event
- **AppInfo_Language** – The language the application is running under
- **AppVersionLong** – The application version
- **Channel** – The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **Reason** - Details on the invalid entry in preferences
- **SessionId** - The identifier for the session

appdelegate.launch

This event denotes that an attempt to launch the app occurred. We log its result (failure or success). We use this event to identify cases in which MAU fails to launch

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppversionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)

- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - A set of static text indicating launch status.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

appdelegate.terminate

This event denotes that a graceful Application Exit occurred. We use this event to distinguish Graceful Application Exits from ungraceful ones.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wifi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Static text indicating Microsoft Autoupdate has terminated.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

appinstall.connecttoxpc

This event denotes that errors occurred connecting to MAU Helper (a component that performs application installation). This event denotes a potential corruption of the MAU application. The device will not be able to install updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains error information on the connection issue.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

appinstall.logscanned

This event is used to determine if log file was successfully processed. We use this event to detect and address any issues arise during application installation.

The following fields are collected:

- **App** – The application process sending the event
- **AppInfo_Language** – The language the application is running under
- **AppVersionLong** – The application version
- **Channel** – The preference for audience
- **Device_NetworkCountry** – The device country (based on IP address)
- **DeviceID** – The device identifier
- **DeviceInfo_Model** – The hardware model of the device
- **DeviceInfo_NetworkType** – The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** – The version of the operating system
- **Event_ReceivedTime** – The time at which telemetry was received

- **EventInfo_Name** – The name of the telemetry event being logged
- **EventInfo_Time** – The time at which the logged event took place
- **HowTocheck** – The preference for checking of updates
- **Payload** – Reports on errors found during application install and/or scan completion status
- **PipelineInfo_ClientCountry** – The device country (based on IP address)
- **PipelineInfo_ClientIp** – The first three octets of the IP address
- **SessionId** – The identifier for the session

appinstall.xpcremoteobjecterror

This event reports on an error found while attempting to connect to Privileged Helper Tool via XPC connection. We use this event to track and address possible MAU installation issues.

The following fields are collected:

- **App** – The application process sending the event
- **AppID** – The application identifier.
- **AppInfo_Language** – The language the application is running under
- **AppVersionLong** – The application version
- **Channel** – The preference for audience
- **Device_NetworkCountry** – The device country (based on IP address)
- **DeviceID** – The device identifier
- **DeviceInfo_Model** – The hardware model of the device
- **DeviceInfo_NetworkType** – The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** – The version of the operating system
- **Event_ReceivedTime** – The time at which telemetry was received
- **EventInfo_Name** – The name of the telemetry event being logged
- **EventInfo_Time** – The time at which the logged event took place
- **HowTocheck** – The preference for checking of updates
- **Payload** – Contains information on the nature of proxy error encountered
- **PipelineInfo_ClientCountry** – The device country (based on IP address)
- **PipelineInfo_ClientIp** – The first three octets of the IP address
- **SessionId** – The identifier for the session

appregistry.config

This event reports on any errors encountered while loading application registry information. We use this report to advise IT Admins on the correct format of setting up client application registrations.

The following fields are collected:

- **App** – The application process sending the event
- **AppInfo_Language** – The language the application is running under

- **AppVersionLong** – The application version
- **Channel** – The preference for audience
- **Device_NetworkCountry** – The device country (based on IP address)
- **DeviceID** – The device identifier
- **DeviceInfo_Model** – The hardware model of the device
- **DeviceInfo_NetworkType** – The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** – The version of the operating system
- **Event_ReceivedTime** – The time at which telemetry was received
- **EventInfo_Name** – The name of the telemetry event being logged
- **EventInfo_Time** – The time at which the logged event took place
- **HowTocheck** – The preference for checking of updates
- **Payload** – Contains information on the nature of error encountered with application registration.
- **PipelineInfo_ClientCountry** – The device country (based on IP address)
- **PipelineInfo_ClientIp** – The first three octets of the IP address
- **SessionId** – The identifier for the session

appregistry.info

This event denotes that the application launched. We use this event to list the applications for which MAU can control updates, the number of copies available as well as their version and install location (default or other).

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains information on list of identifiers application uses to register with Microsoft Autoupdate services and number of installations registered for the application.

- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

appregistry.remove

This event denotes that an attempt to remove an App from the list of applications Managed by MAU took place. We use this event to confirm that only MAU-released applications are managed via MAU (no AppStore apps should appear here).

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Name and identifier of the application being removed, whether the application still exists in the registered location and if the application was installed from AppStore.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

catalog.errorsignature

This event reports on various problems with downloaded files, including vendor signature and hash value mismatch on downloaded file. We use this event to detect problems in publishing manifest set for applications.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)

- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **FileHash** – Hash value of the downloaded file
- **FileName** – The name of the file showing hash value mismatch
- **HashInCatalog** – Hash value entry in the corresponding catalog file
- **HowTocheck** - The preference for checking of updates
- **Payload** - contains information on the app reporting problem
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

catalog.invalid

This event logs an error condition pointing to invalid manifest catalog downloaded. We use this event to ensure no errors are present in published manifest files.

The following fields are collected:

- **App** – The application process sending the event
- **AppInfo_Language** – The language the application is running under
- **AppVersionLong** – The application version
- **CatalogFile** – Name of the catalog file that caused error condition.
- **Channel** – The preference for audience
- **Device_NetworkCountry** – The device county (based on IP address)
- **DeviceID** – The device identifier
- **DeviceInfo_Model** – The hardware model of the device
- **DeviceInfo_NetworkType** – The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** – The version of the operating system
- **Event_ReceivedTime** – The time at which telemetry got received
- **EventInfo_Name** – The name of the telemetry event being logged
- **EventInfo_Time** – The time at which the logged event took place
- **HowTocheck** – The preference for checking of updates
- **PipelineInfo_ClientCountry** – The device country (based on IP address)

- **PipelineInfo_ClientIp** – The first 3 octets of the IP address
- **SessionId** – The identifier for the session

cloningtask.begin

This event indicates start of cloning task prior to application update. We use this event in conjunction with cloningtask.status event to determine volume of cloning failures in order to determine whether cloning feature should be throttled on different audience channels.

The following fields are collected:

- **App** – The application process sending the event
- **AppID** – The application identifier.
- **AppInfo_Language** – The language the application is running under
- **AppVersionLong** – The application version
- **Channel** – The preference for audience
- **Device_NetworkCountry** – The device country (based on IP address)
- **DeviceID** – The device identifier
- **DeviceInfo_Model** – The hardware model of the device
- **DeviceInfo_NetworkType** – The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** – The version of the operating system
- **Event_ReceivedTime** – The time at which telemetry was received
- **EventInfo_Name** – The name of the telemetry event being logged
- **EventInfo_Time** – The time at which the logged event took place
- **HowTocheck** – The preference for checking of updates
- **PipelineInfo_ClientCountry** – The device country (based on IP address)
- **PipelineInfo_ClientIp** – The first three octets of the IP address
- **SessionId** – The identifier for the session
- **UpdateID** – The identifier for update.

cloningtask.helpertoolconnection

This event records issues with install on clone (i.e. either we fail to connect to the helper to apply an update, or we connect but the helper is unable to apply the update). If we ever get a record reported, this means install on clone has failed and will now have to fall back to an in-place update.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)

- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an ID to identify a single update activity, and Proxy Error reported during cloning process.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

cloningtask.status

This event indicates status of cloning process for the application to be updated. We use this event to determine success rate as well as types of errors encountered causing failures. This event is used to determine whether cloning feature should be throttled on different audience channels.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - The application identifier.
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Error** - The string contains error information if error occurred during cloning task.
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address

- **SessionId** - The identifier for the session
- **Success** - The string representation of a Boolean variable.
- **UpdateID** - The identifier for update.

cloningtask.status.finish

This event reports on the completion of "cloning" task. This event forms part of the update funnel report and we use it to determine health of application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - The application identifier
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **Success** - Indication whether the cloning task succeeded
- **UpdateID** - The update identifier.

configuration.channel

This event records attempts to switch Channels (Audience Group) in MAU. We use this to log attempts and their results (success or failure).

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier

- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains selected Channel Name.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

configuration.metadata

This event is logged whenever MAU is initializing. It is a MAU heartbeat type of event

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Static text indicating either individual metadata is being initialized, or configuration is being initialized.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session.

configuration.systemVersion

This event indicates an attempt to retrieve system version has failed. This also contains information on the information Microsoft Auto Update (MAU) was able to collect from the system. We use this event to determine

whether MAU should cater for failures. Note that system version is used to determine whether an update can be applied to the client device.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains information on error encountered while retrieving macOS system version string.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

controller.alertmanager.reinstallresponse

This event denotes that MAU fell into an unusable/unrecoverable state and needs to be reinstalled. This event denotes an unrecoverable error and user intervention is required.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received

- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference of checking for updates
- **Payload** - Contains enumerated user selection.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.alertmanager.tmpdiskfull

This event denotes that insufficient disk space was detected. We will not be able to install updates due to insufficient disk space.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Static text.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.alertmanager.tmpdiskfullretry

This event denotes that a retry attempt to install an update was initiated after Insufficient disk space was detected. We retry the installation after not being able to install updates due to insufficient disk space.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version

- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Static text.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.alertmanager.tmpdiskfullretrycancel

This event denotes that a cancellation on an install-retry attempt after Insufficient disk space was detected. We use this event to determine if our fallback mechanism was enough to guide the user thru the update process when insufficient disk was detected.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Static text.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)

- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.checkwindow.noupdatefoundok

This event denotes that a check for updates resulted in no updates found. We use this event for ensuring updates are offered correctly, optimizing service loads, and define how frequent our updates checks should be. We also want to optimize our release cadence based on user expectation of updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Static text.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.checkwindow.updatecheck

This event denotes that a check for updates was performed. We use this event for ensuring updates are offered correctly, optimizing service loads, and define how frequent our updates checks should be. We also want to optimize our release cadence based on user expectation of updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier

- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Static text.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.checkwindow.updatecheckcancel

This event denotes that the process of checking for updates was canceled (either by the user or by the system). We use this event for ensuring updates are offered correctly, optimizing service loads, and define how frequent our updates checks should be. We also want to optimize our release cadence based on user expectation of updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Static text.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.checkwindow.updatecheckcanceluser

This event denotes that the process of checking for updates was canceled by the user. We use this event for ensuring updates are offered correctly, optimizing service loads, and define how frequent our updates checks should be. We also want to optimize our release cadence based on user expectation of updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Static text.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.checkwindow.updatesfound

This event denotes that the process of checking for updates resulted in updates found. We use this event for ensuring updates are offered correctly.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received

- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Static text.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.checkwindow.uptodate

This event denotes that the process of checking for updates resulted no updates because apps in the device are up to date. We use this event for ensuring updates are offered correctly.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Static text.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.applaunchwithpendingupdate

This event denotes that an app that is in the process of getting updates was launched. We use this event for ensuring updates are offered correctly. We should prevent opened apps from getting updates. Apps must be closed prior to update.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under

- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Static text.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.closeapplicationdialog

This event denotes that an app that is in the process of getting updates was launched. We use this event for ensuring updates are offered correctly. We should prevent opened apps from getting updates. Apps must be closed prior to update.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Static text.

- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.curtasknull

This event denotes that an unexpected error occurred while attempting to apply an update. We use this event for ensuring updates are offered correctly.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Static text.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.downloadcancel

This event denotes that the download process was canceled by user. We use this event for ensuring updates are offered correctly.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device

- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Static text.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.downloadfailed

This event denotes that a failure occurred when downloading an update. We use this event for ensuring updates are offered and downloaded correctly.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.downloadfailedok

This event denotes that a failure occurred when downloading an update and the user was notified. We use this event for ensuring updates are offered and download correctly, and that in case of failure, a notification is served to the user.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.downloadpathmissing

This event denotes that a failure occurred when downloading an update. We use this event for ensuring updates are offered and downloaded correctly. This event indicates a download URL is missing.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place

- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.downloadtasknull

This event denotes that a failure occurred when downloading an update. We use this event for ensuring updates are offered and downloaded correctly. This event indicates that Microsoft Autoupdate was asked to pause/resume a download but could not find corresponding download manager.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.filesignaturenotverified

This event denotes that a failure occurred when downloading an update. This event indicates that Microsoft Autoupdate was unable to verify that this update was published by Microsoft. We use this event for ensuring updates are offered and downloaded correctly.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version

- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that contains download URL. This is a Microsoft download location except when the channel is set to Custom. For Custom channel, this value is set to "Custom Location".
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.installcomplete

This event denotes that the installation of all updates offered by Microsoft Autoupdate completed. We use this event for ensuring updates are offered and downloaded correctly.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)

- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.networkunavailablealert

This event denotes that network connectivity was lost while downloading updates. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.networkunavailablealertok

This event denotes that network connectivity was lost while downloading updates. It also denotes that the user was notified of this error. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device

- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.noconnectionok

This event denotes that network connectivity was lost while downloading updates. It also denotes that the user was notified of this error. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.repairrequired

This event denotes that the update process failed. It also denotes that an update was completed but Microsoft Autoupdate found a problem with updated application and repair is required. We use this event for ensuring the

update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.updateaborted

This event denotes that the update process was aborted. It also denotes that an update was already in progress by Daemon and user clicked OK to abort download. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received

- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.updatefailed

This event denotes that one or more updates from the current batch failed. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.updatesuccessful

This event denotes that all updates from the current batch were successful. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version

- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.downloadwindow.userpaused

This event denotes that all updates from the current batch were successful. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address

- **SessionId** - The identifier for the session

controller.downloadwindow.userresumed

This event denotes that the download updates process was resumed successfully after going into pause. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.mainwindow.setautomaticchecking

This event denotes that the device was enrolled into Automatic Update mode. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** – The application process sending the event
- **AppInfo_Language** – The language the application is running under
- **AppVersionLong** – The application Version
- **Channel** – The preference for audience
- **Device_NetworkCountry** – The device country (based on IP address)
- **DeviceID** – The device identifier
- **DeviceInfo_Model** – The hardware model of the device
- **DeviceInfo_NetworkType** – The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** – The Version of the Operating System

- **Event_ReceivedTime** – The time at which telemetry was received
- **EventInfo_Name** – The name of the telemetry event being logged
- **EventInfo_Time** – The time at which the logged event took place
- **HowTocheck** – The preference for checking of updates
- **Payload** – Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** – The device country (based on IP address)
- **PipelineInfo_ClientIp** – The first 3 octets of the IP address
- **SessionId** – The identifier for the session

controller.mainwindow.setautomaticdownloadinstall

This event denotes that the device was enrolled into Automatic Update mode. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.mainwindow.setmanualchecking

This event denotes that the device was enrolled into Manual Update mode. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under

- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.templateawindow.cancel

This event denotes that the user chose to cancel or ignore a provided warning message. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)

- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.templateawindow.enroll

This event denotes that the user chose to follow a provided warning recommendation. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.templateawindow.install

This event denotes that the user chose to follow a provided warning recommendation related to initiating a software installation action. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device

- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.updatewindow.begindownloadingapps

This event denotes that the download for updates was started via Update Window. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains a dictionary of available update packages and an indication whether user selected to install that entry.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.updatewindow.networkretry

This event denotes that a retry was triggered at the Update Sheet due to network failure. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.updatewindow.networkretrycancel

This event denotes that a retry could not be triggered at the Update Sheet due to network failure. This event indicates user elected to cancel updates after being alerted of network becoming unavailable. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged

- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.updatewindow.networkunavailable

This event denotes that network connectivity was suddenly lost. This event indicates server is not reachable when attempting to download an update package. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.updatewindow.noupdateavailable

This event denotes that there was a search for updates that resulted in no updates being available. This event indicates no available updates were found by Microsoft Autoupdate. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under

- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.updatewindow.noupdatestoselect

This event denotes that an error occurred resulting in an empty list of updates. This event indicates Microsoft Autoupdate is showing an empty update sheet. This should not happen. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.

- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

Controller.UpdateWindow.UpdateAvailable

This event denotes that there was a search for updates that resulted in updates being offered. We use this even to determine whether updates are being offered for the user to see, whether the proper updates are being shown, or whether update blocking is working as expected. We use this event to ensure the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains a dictionary of available update packages and user selection status for each.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

controller.updatewindow.updateavailablecancel

This event denotes that a user canceled after we displayed the update sheet listing updates. We use this even to explain reasons for not updating (i.e. user willingly cancels). We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)

- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadactor.pause

This event denotes that the user issued a request to pause the download. We use this even to explain reasons for updates apparently not completing. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadactor.redirect

This event denotes that the downloader agent is pointed to an endpoint that issues a URL redirect for the download request. We use this even to explain reasons for download failure and diagnose proxy issues. It can also help diagnose reasons why users are observed to install older builds. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains redirected URL. This is Microsoft download location except when the channel is set to Custom. For Custom channel, this value is set to "Custom Location".
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadactor.resume

This event denotes the user issues a request to resume a paused download. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)

- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadactor.resumeerror

This event denotes the user issues a request to resume a paused download. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains download URL path. This is Microsoft download location except when the channel is set to Custom. For Custom channel, this value is set to "Custom Location".
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadactor.status

This event logs that there are attempts to fetch collateral files and their result (Success or Failure). We want to know the collaterals and packages being fetched. A wrong file being fetched can indicate a build/collateral issue. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains download URL, and an error code in case of failure. Download URL is Microsoft download location except when the channel is set to Custom. For Custom channel, this value is set to "Custom Location".
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadmanifest.configuration

This event reports an error with Microsoft Auto Update (MAU) configuration - either with Custom Server setup in preferences or in endpoint definitions in Update Assistant in installed MAU components. We use this event to advise IT Admins to set correct Manifest Server endpoints.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received

- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **Payload** -Indicates whether error lies with custom server setup, or installed MAU components
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

downloadmanifest.downloadcatalogfail

This event a download failure occurred. The file that failed to download is logged. We want to know the collaterals and packages being fetched. A failure to download a manifest can point to either a build collateral generation failure, a CDN configuration error, a client configuration error, a network error. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains download error code and a download file URL. This is Microsoft download location except when the channel is set to Custom. For Custom channel, this value is set to "Custom Location".
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadmanifest.downloadcatalogsuccess

This event denotes that a file was successfully downloaded. A failure to download a manifest can point to either a build collateral generation failure, a CDN configuration error, a client configuration error, a network error. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event

- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains download error code and a download file URL. This is Microsoft download location except when the channel is set to Custom. For Custom channel, this value is set to "Custom Location".
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadmanifest.downloadfail

This event denotes that a download error occurred. The manifest or package file that failed to download as well as error details are logged. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates

- **Payload** - Contains download error code and a download file URL. This is Microsoft download location except when the channel is set to Custom. For Custom channel, this value is set to "Custom Location".
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadmanifest.downloadfromurl

This event denotes that the downloading of a catalog file has started. We log the URL from which the catalog file is being downloaded. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains download error code and a download file URL. This is Microsoft download location except when the channel is set to Custom. For Custom channel, this value is set to "Custom Location".
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadmanifest.downloading

This event denotes that the downloading of a catalog file has started. We log the URL from which the catalog file is being downloaded. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version

- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains download error code and a download file URL. This is Microsoft download location except when the channel is set to Custom. For Custom channel, this value is set to "Custom Location".
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadmanifest.downloadsuccess

This event denotes that the downloading of an XML and package file has succeeded. We log the URL from which the file is being downloaded. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains download error code and a download file URL. This is Microsoft download location except when the channel is set to Custom. For Custom channel, this value is set to "Custom Location".

- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadmanifest.downloadurl

This event denotes that a request to download a file occurred. We log the URL from which the file is being downloaded. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains download error code and a download file URL. This is Microsoft download location except when the channel is set to Custom. For Custom channel, this value is set to "Custom Location".
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadmanifest.filenameerror

This event denotes that an unexpected error occurred. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)

- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadmanifest.invalidhash

This event denotes a security validation of our files failed. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains name of the downloaded file with invalid hash value.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadmanifest.missingdaemon

This event denotes a user attempted to check for updates and we discovered that MAU was missing a core component (daemon). We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadmanifest.signatureerror

This event denotes that the code signature verification failed for a package. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received

- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains a name of the downloaded file with invalid hash value.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadmanifest.status

This event logs a summarized aggregation of attempts/failures hit during the download process for manifest and package files. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains information including URL (Microsoft address), prefix of the file being downloaded, any errors encountered, etc.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadmgr.downloadend

This event logs a marker that indicates the download process completed on its own. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event

- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains information including URL (Microsoft address), prefix of the file being downloaded, any errors encountered, etc.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadmgr.downloadstart

This event logs the update that is about to be downloaded. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates

- **Payload** - Contains the name of the update being downloaded.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

downloadtask.downloadbegin

This event indicates start of download activity for an application update. This forms part of update funnel and we use this to determine health of application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **BundleVersion** - Version of the application being updated
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **PreviousUpdateID** - Identifier for an application update
- **SessionId** - The identifier for the session
- **UpdateID** - Identifier for an application update
- **UpdatePkg** - Name of the update package being applied
- **UpdateVersion** - Version of the application after the update

downloadtask.downloadfailure

This event logs that an error occurred downloading a package file. We log the update path and the error. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** – The identifier of application that has download failure.

- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Error** – The error observed during download.
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains the name of the update being downloaded and the error observed. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** – The identifier of the update being downloaded.

downloadtask.downloadsuccess

The successful downloading a package file. We log the update path used. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** – The identifier of application.
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received

- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains the update path for the successful download.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** – The identifier of downloaded update.

downloadtask.updatertypeerror

This event reports on an updater type error in the downloaded manifest file. We use this event to notify owner of the manifest file so that the error can be corrected.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** - Identifier for an application update
- **UpdaterType** - Type of updater specified in the downloaded manifest file
- **UpdateURL** - URL of update package that needs to be applied

downloadtask.urlerror

This event reports on an error in the URL specified in downloaded manifest file. We use this event to notify owner of the manifest file so that the error can be corrected.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **Error** - Indicates nature of error being encountered
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** - Identifier for an application update
- **UpdateURL** - URL of update package that needs to be applied

fba.changelastupdate

This event reports on when Microsoft Auto Update (MAU) has checked for updates. We use this event to debug when a particular device has not been offered an update for an extended period of time.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, Unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place

- **Payload** - Contains date time of when MAU checked for updates
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

fba.checkforupdate

This event denotes that a background process is checking for updates. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fba.checkforupdateskip

This event denotes that a background process skipped update due to MAU GUI being opened. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier

- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fba.forceinstallmsgsent

This event indicates that a forced update is initiated from the user interface. This event forms part of funnel and is used to determine health of force update feature.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

fba.forceupdatecheck

This event indicates update check is forced. We use this event to determine volume of forced update checks which happen outside normal update check cycle.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

fba.guiappopen

This event indicates that the user interface is being launched under Automatic Check mode, as an application with applicable update is currently open. This event is used to determine volume of user interface launches from Automatic Check mode for future feature development.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place

- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

fba.installpending

This event indicates Microsoft Auto Update (MAU) sent a notification regarding pending updates. This event is used to determine volume of updates that are initiated from user notifications and is used to enhance user experience by minimizing interruption to the user in the future releases.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

fba.launch

This event indicates start of Microsoft Update Assistant with the method of being launched. This event is used to determine whether Microsoft Update Assistant is being launched in incorrect context.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience

- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

fba.launchbyagent

This event indicates that the Microsoft Update Assistant has been launched via Launch Agent. This event is used to determine volume of Microsoft Update Assistant being launched from user interface for future development.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

fba.launchfromprotocol

This event indicates that the Microsoft Update Assistant has been launched via URL protocol. This event is used to determine volume of Microsoft Update Assistant being launched via URL for future development.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Contains information about URL scheme and URL Host
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

fba.launchgui

This event indicates that the Microsoft Update Assistant is attempting to launch Graphical User Interface (GUI). This event is used to determine volume of UI launches initiated from Microsoft Update Assistant, to help with future development, including minimizing user interruption from frequent UI launch.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system

- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

fba.launchstatus

This event logs MAU's daemon failures while attempting to launch. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Error** – Contains OSStatus (Apple status code) reflecting launch status.
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains OSStatus (Apple status code) reflecting launch status. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session
- **Success** – The string Boolean indicating whether MAU daemon process was successfully launched.

fba.mausilentupdate

This event indicates Microsoft Update Assistant is initiating silent updates. This event is used to determine volume of updates that are applied without user intervention, to help drive enhancements in user experience.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **Reason** - Static text indicating a silent update cannot proceed as UI is open
- **SessionId** - The identifier for the session

fba.moreinfofromappnotification

This event reports on information that a registered application is routing through Microsoft Auto Update (MAU). For example, end-of-service messages are pushed through MAU notification. We use this event to determine the volume of devices that are displaying this particular notification, to determine the success of information dissemination.

The following fields are collected:

- **AdditionalInfoID** - Uniquely identifies “More Info” being pushed through MAU.
- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)

- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **NotificationEvent** - Static text indicating what type of notification is being applied.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

fba.multipledaemon

This event indicates that another instance of Microsoft Update Assistant has been detected and current instance will be terminated. We will be using this event to determine volume of devices that attempt to run multiple instances of Update Assistant and design a workaround if need arises.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

fba.nofifyappclosed

This event indicates that the Microsoft Update Assistant is sending a notification for pending updates because there are no registered applications open and updates can proceed without interrupting the user. We use this event to determine volume of updates that can be applied but need user action to do so. This event is used to help drive enhancement in user experience.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

fba.nofifyappopen

This event indicates that the Microsoft Update Assistant is sending a notification for pending updates because there are registered applications open and updates will require the applications closed to proceed. We use this event to determine volume of updates that require user intervention. This event is used to help drive enhancement in user experience.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged

- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

fba.settimerfail

This event indicates an attempt to set up timer to trigger a future update has failed. This event is critical, and we use this event to determine volume of failures to develop work-around if needed.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Contains information about last update time and calendar being used
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

fba.silentupdateoptin

This event denotes that the user is Opting into silent updates. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience

- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fba.skipforcedupdate

This event indicates forced update check is being skipped due to open applications. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fba.startforcedupdate

This event indicates that an attempt to apply a forced update has occurred. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fba.terminate

This event indicates that the MAU daemon terminated normally. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received

- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fba.updatefound

This event indicates that the MAU daemon has found available updates to offer. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains number of available updates found.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fba.updatetimer

This event indicates Microsoft Autoupdate Daemon process became active to check for updates after sleeping for set amount of time. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under

- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains current date time information.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fbasilentupdate.allappsclosed

This event logs if all apps were closed prior to an install. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)

- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fbasilentupdate.applaunchafterupdate

This event logs an attempt to relaunch the app after a silent update and the update mode (clone or not). We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** – The identifier of the application.
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Error** – The detail of the error occurred during launching application after update.
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity and name of the application be launched. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fbasilentupdate.applaunchwileinstalling

We log when an app launch was made while installing an update. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience

- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fbasilentupdate.appneedtoclose

We log when an update process was kicked off and we find that the application to update was opened. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity, name of an update and application bundle ID.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address

- **SessionId** - The identifier for the session

fbasilentupdate.appterminationeventreceived

This event indicates Microsoft Autoupdate has received an Apple event informing the application has been terminated. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** – The identifier of the application.
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Error** – The detail about error occurred during application termination.
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity and application bundle ID. This may also contain an error string if Microsoft Autoupdate determines the application is still running even though termination event was received. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** – The identifier of the application update.

FBASilentUpdate.ClientSession

This event is used to calculated critical update health metric for Microsoft Auto Update (MAU). This event allows us to indicate which update session (download or install) the backend is currently handling.

The following fields are collected:

- **App** – The application process sending the event
- **AppInfo_Language** – The language the application is running under
- **AppVersionLong** – The application version

- **Channel** – The preference for audience
- **Device_NetworkCountry** – The device country (based on IP address)
- **DeviceID** – The device identifier
- **DeviceInfo_Model** – The hardware model of the device
- **DeviceInfo_NetworkType** – The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** – The version of the operating system
- **Event_ReceivedTime** – The time at which telemetry was received
- **EventInfo_Name** – The name of the telemetry event being logged
- **EventInfo_Time** – The time at which the logged event took place
- **HowTocheck** – The preference for checking of updates
- **Payload** – Indicates which update session (download or install) the backend is currently handling.
- **PipelineInfo_ClientCountry** – The device country (based on IP address)
- **PipelineInfo_ClientIp** – The first three octets of the IP address
- **SessionId** – The identifier for the session

fbasilentupdate.codesignfailure

This event logs the result of codesign verification after applying an update. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains the result of the codesign verification operation.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address

- **SessionId** - The identifier for the session

fbasilentupdate.download

This event denotes that an update is being downloaded. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity and name of an update.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **ScreenLocked** – Indication whether download is initiated behind locked screen
- **SessionId** - The identifier for the session

fbasilentupdate.downloadfailed

This event denotes that a failure occurred while downloading an update. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** – The identifier of the application.
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device

- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Error** – The detail of error occurred during application update download.
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity and name of an update. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** – The identifier of the application update.
- **UpdateName** – The name of the application update.

fbasilentupdate.downloadinbackground

This event denotes that we are starting the download a set of updates in the background (we log the number of updates being concurrently downloaded). We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains number of updates queued.
    - **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address

- **SessionId** - The identifier for the session

fbasilentupdate.downloadingrepairupdate

This event denotes that we have initiated an attempt to download a repair for a failed update. We log the version and the update. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity and name of an update.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **ScreenLocked** – Indication whether download is initiated behind locked screen
- **SessionId** - The identifier for the session

fbasilentupdate.duplicatedownloadattempted

This event denotes that an error occurred. We should only be downloading one update for a given app at a time. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device

- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fbasilentupdate.installattemptfailed

This event denotes that an installation attempt for an update (version) failed. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fbasilentupdate.installcomplete

This event denotes that all updates on the batch finished installing. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fbasilentupdate.installed

This event denotes that an individual update was installed successfully. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates

- **Payload** - Text that indicates the nature of the event. Contains the update identifier.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fbasilentupdate.installing

This event denotes that an individual update was initiated. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity, name of an update and update package name.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fbasilentupdate.installstatus

This event reports on the status of the application update task. This event forms part of the application update funnel and we use it to monitor the health of application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience

- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Contains information whether progress view is shown
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **Success** - Indication whether application update was successful
- **UpdateID** - Identifier for an application update
- **UpdateName** - Name of the update as it appears in the downloaded manifest file
- **UpdatePkg** - Name of the update package being applied

fbasilentupdate.notificationerror

This event reports on an error encountered while trying to send user notification. This event will be used to debug cause of error and take corrective actions.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **ErrType** - Indicates nature of error encountered
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place

- **HowToCheck** - How to check setting
- **Message** - Content of notification
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **Title** - Title of the notification
- **Type** - Type of notification

fbasilentupdate.notificationremoved

This event denotes that an update that was blocked is no longer blocked. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an application ID (identifier application uses to register with Microsoft Autoupdate service) for the previously blocked application
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fbasilentupdate.queueinstall

This event denotes that an update will be queued for silent installation. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under

- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity and name of the update.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fbasilentupdate.requiredappsclosed

We log when an application that has a pending update has been closed. This indicates the time at which the actual install can proceed. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity, application bundle ID.

- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

FBASilentUpdate.TimerForAppTermination

This event is used to calculated critical update health metric for Microsoft Auto Update (MAU). This event allows us to keep track of the termination event of the opened application and the duration of its opened state.

The following fields are collected:

- **App** – The application process sending the event
- **AppInfo_Language** – The language the application is running under
- **AppVersionLong** – The application version
- **Channel** – The preference for audience
- **Device_NetworkCountry** – The device country (based on IP address)
- **DeviceID** – The device identifier
- **DeviceInfo_Model** – The hardware model of the device
- **DeviceInfo_NetworkType** – The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** – The version of the operating system
- **Event_ReceivedTime** – The time at which telemetry was received
- **EventInfo_Name** – The name of the telemetry event being logged
- **EventInfo_Time** – The time at which the logged event took place
- **HowTocheck** – The preference for checking of updates
- **Payload** – Indicates whether a timer was set for an open application when its update installation was triggered.
- **PipelineInfo_ClientCountry** – The device country (based on IP address)
- **PipelineInfo_ClientIp** – The first three octets of the IP address
- **SessionId** – The identifier for the session

fbasilentupdate.updateavailablenotification

This event indicates an update available notification is triggered. We must ensure the flow to prompt for updates is triggered when an update is detected. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **CustomNotification** – Boolean indicating whether custom notification was used.

- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fbasilentupdate.userclicknotification

This event indicates user clicked content section of the update available notification and Microsoft Autoupdate GUI is being launched. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address

- **SessionId** - The identifier for the session

fbasilentupdate.userselectinstalllater

This event indicates user opted to install later after shown update available notification. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

fbasilentupdate.userselectinstallnow

This event indicates user opted to install now after shown update available notification. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System

- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

gui.dashboardrowview.updatestate

This event reports on an error found while attempting to display application information in MAU UI. We use this event to ensure health of MAU and track and address failures.

The following fields are collected:

- **App** – The application process sending the event
- **AppID** – The application identifier.
- **AppInfo_Language** – The language the application is running under
- **AppVersionLong** – The application version
- **Channel** – The preference for audience
- **Device_NetworkCountry** – The device country (based on IP address)
- **DeviceID** – The device identifier
- **DeviceInfo_Model** – The hardware model of the device
- **DeviceInfo_NetworkType** – The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** – The version of the operating system
- **Event_ReceivedTime** – The time at which telemetry was received
- **EventInfo_Name** – The name of the telemetry event being logged
- **EventInfo_Time** – The time at which the logged event took place
- **HowTocheck** – The preference for checking of updates
- **Payload** – Contains information on the nature of error encountered
- **PipelineInfo_ClientCountry** – The device country (based on IP address)
- **PipelineInfo_ClientIp** – The first three octets of the IP address
- **SessionId** – The identifier for the session

gui.dashboardview.appisopendialog.display

This event indicates that the UI has shown a dialog to close an open application to proceed with application update. This event is used to determine volume of updates being delayed in order to provide future enhancements to minimize user interruption.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** - Identifier for an application update
- **UpdateName** - Name of the update as it appears in the downloaded manifest file

gui.dashboardview.appisopendialogbutton.clicked

This event indicates whether application update is skipped, or another attempt is being made after showing an open application dialog. This event is used to determine volume of updates that are being skipped and used for future enhancements to minimize user interruption.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **ButtonType** - Skip or Retry
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)

- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** - Identifier for an application update
- **UpdateName** - Name of the update as it appears in the downloaded manifest file

gui.dashboardview.updateinprogressdialog.display

This event logs whether a dialog was displayed to users indicating update is already in progress.

The following fields are collected:

- **App** – The application process sending the event
- **AppInfo_Language** – The language the application is running under
- **AppVersionLong** – The application version
- **Channel** – The preference for audience
- **Device_NetworkCountry** – The device country (based on IP address)
- **DeviceID** – The device identifier
- **DeviceInfo_Model** – The hardware model of the device
- **DeviceInfo_NetworkType** – The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** – The version of the operating system
- **Event_ReceivedTime** – The time at which telemetry was received
- **EventInfo_Name** – The name of the telemetry event being logged
- **EventInfo_Time** – The time at which the logged event took place
- **HowTocheck** – The preference for checking of updates
- **PipelineInfo_ClientCountry** – The device country (based on IP address)
- **PipelineInfo_ClientIp** – The first three octets of the IP address
- **SessionId** – The identifier for the session

gui.dashboardview.updatemodebutton.clicked

This event indicates update mode changed from UI control. This event is used to determine volume of devices that transition from one mode to another and is used to help determine why customers are moving away from automatic updates.

The following fields are collected:

- **App** - The application process sending the event

- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Indication whether automatic download is turned OFF
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

gui.feedbackwindow.buttonclicked

This event reports on whether feedback is submitted or canceled before submission. This event is used to help determine volume of feedback sent on a particular release version. This helps to isolate potential issues early.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **ButtonType** - Indication whether feedback is sent or canceled
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting

- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

gui.preferenceview.consentsheet.display

This event indicates that a consent sheet for a given channel is displayed, if available. This event is used to determine volume of devices that newly enroll into applicable audience channel (Insider Fast / Insider Slow). We also use this event to ensure consent dialog display is functioning to show terms of usage to users.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **ChannelName** - Channel for which the consent dialog is displayed
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

gui.preferenceview.consentsheet.licenseerror

This event reports on the error encountered while attempting to show consent dialog. This event is critical and is used to correct any issues caused by a change in product, if applicable.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier

- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **ErrorCode** - Error code encountered
- **ErrorDomain** - Error domain
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

gui.preferenceview.switchchannel

This event reports on transition between user selected channels. This event is used to help determine why customers are opting out of Insiders channels.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **PickedFrom** - Old channel
- **PickedTo** - New channel
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

gui.updatemanager.applaunchduringupdate

This event reports that an application was launched while it was being updated, and Microsoft AutoUpdate is terminating the launched application. Note that launching an application while being updated may result in application corruption. We use this event to ensure update process is not impacted by launched application before it is ready to be used.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - The identifier of application that was launched during updates.
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **Success** - The string Boolean value indicating whether application was successfully terminated.
- **UpdateID** - The identifier of application update.

gui.updatemanager.downloadupdateforapp

This event reports on download completion status for an update. We use this event to ensure health of update process and track/address failure point.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - The application identifier.
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)

- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **IsRepair** - The string Boolean indicates whether particular update is a repair download.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **isRepair** - Indication whether the download was for a repair download for a previously failed update.
- **UpdateID** - The update identifier.
- **UpdateName** - The update name.

gui.updatemanager.error

This event reports back on any errors encountered during application updates. This may indicate error in Microsoft Auto Update (MAU) execution sequence. We use this report to apply updates to MAU to cater for common error scenarios.

The following fields are collected:

- **App** – The application process sending the event
- **AppInfo_Language** – The language the application is running under
- **AppVersionLong** – The application version
- **Channel** – The preference for audience
- **Device_NetworkCountry** – The device country (based on IP address)
- **DeviceID** – The device identifier
- **DeviceInfo_Model** – The hardware model of the device
- **DeviceInfo_NetworkType** – The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** – The version of the operating system
- **Event_ReceivedTime** – The time at which telemetry was received
- **EventInfo_Name** – The name of the telemetry event being logged
- **EventInfo_Time** – The time at which the logged event took place
- **HowTocheck** – The preference for checking of updates
- **Payload** – Contains information on error encountered during an application update.

- **PipelineInfo_ClientCountry** – The device country (based on IP address)
- **PipelineInfo_ClientIp** – The first three octets of the IP address
- **SessionId** – The identifier for the session
- **Success** – The string Boolean value indicating whether application was successfully terminated.

gui.updatemanager.installcleanupforapp

This event indicates temporary files created during application installation were successfully cleaned up. This forms part of update funnel used to determine application update health.

The following fields are collected:

- **App** – The application process sending the event
- **AppID** – The application identifier.
- **AppInfo_Language** – The language the application is running under
- **AppState** – The integer indicates the state of application after update attempt.
- **AppVersionLong** – The application version
- **Channel** – The preference for audience
- **Device_NetworkCountry** – The device country (based on IP address)
- **DeviceID** – The device identifier
- **DeviceInfo_Model** – The hardware model of the device
- **DeviceInfo_NetworkType** – The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** – The version of the operating system
- **Event_ReceivedTime** – The time at which telemetry was received
- **EventInfo_Name** – The name of the telemetry event being logged
- **EventInfo_Time** – The time at which the logged event took place
- **HowTocheck** – The preference for checking of updates
- **PipelineInfo_ClientCountry** – The device country (based on IP address)
- **PipelineInfo_ClientIp** – The first three octets of the IP address
- **SessionId** – The identifier for the session
- **UpdateID** – The update identifier.

gui.updatemanager.installsuccessforapp

This event indicates successful application update. This event forms part of the update funnel which we use to determine update health.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - The application identifier.
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version

- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **Success** - The string Boolean indicates whether updates were successfully installed.
- **UpdateID** - The update identifier.

gui.updatemanager.installupdateforapp

This event indicates start of actual install process for an application update. This event forms part of application update funnel which we use to determine update health.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - The application identifier.
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **PipelineInfo_ClientCountry** - The device country (based on IP address)

- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** - The update identifier.

gui.updatemanager.queueinstallforapp

This event indicates start of actual install process for an application update. This event forms part of application update funnel which we use to determine update health.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - The application identifier.
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** - The update identifier.

gui.updatemanager.relaunchapp

This event logs whether applications were successfully relaunched after updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - The application identifier.
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)

- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **Success** - The string Boolean value indicating whether application was successfully terminated.
- **UpdateID** - The update identifier.
- **UpdateName** - The update name.

installdata.checkrunning

This event logs the result of a check between the apps to be installed and whether the installation attempt will proceed based on the app being opened. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)

- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installdata.cleanup

Package files should be removed after installation. This event records instances in which we fail to remove them. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains downloaded file name and error details.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installedapp.acknowledgedcoreappleevent

This event indicates Microsoft Auto Update (MAU) has received an Apple event acknowledgment from a registered application to terminate the application to proceed with pending application update. This event is used to help develop future enhancement to minimize user interruption during application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppleEventClass** - Indicates type of event being sent/acknowledged
- **AppleEventID** - Unique identifier for the event being sent/acknowledged
- **AppVersionLong** - The application version
- **Channel** - The preference for audience

- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Contains retry count
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** - The update identifier

installedapp.invalidbundle

This event indicates Microsoft Autoupdate could not retrieve bundle information for the registered application at given path. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains application name.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)

- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installedapp.invalidpreference

This event logs cases in which the user preference contains an invalid application entry. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference of checking for updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installedapp.nilbundleid

This event logs cases in which the bundle ID was missing for an app. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)

- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains application name.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installedapp.nilbundlename

This event logs cases in which the bundle name was missing for an app. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains application name.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installedapp.respondedcoreappleevent

This event indicates Microsoft Auto Update (MAU) has received an Apple event response code from a registered application to terminate the application in order to proceed with pending application update. This event is used to help develop future enhancement to minimize user interruption during application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppleEventClass** - Indicates type of event being sent/acknowledged
- **AppleEventID** - Unique identifier for the event being sent/acknowledged
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Contains retry count
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** - The update identifier

installedapp.sendcoreappleevent

This event indicates Microsoft Auto Update (MAU) is sending an Apple event to a registered application to terminate the application in order to proceed with pending application update. This event is currently being used to help develop future enhancement to minimize user interruption during application updates.

The following fields are collected:

- **Acknowledged** - Indicates whether the subject application has acknowledged receipt of the event
- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppleEventClass** - Indicates type of event being sent/acknowledged
- **AppleEventID** - Unique identifier for the event being sent/acknowledged
- **AppVersionLong** - The application version
- **Channel** - The preference for audience

- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Contains retry count
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **Success** - Indicates whether the subject application has reported success of operation
- **UpdateID** - The update identifier

installstatus.codesign

This event logs the status of the OS codesign binary. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)

- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installstatus.daemon

This event logs the status of the Microsoft AutoUpdate daemon. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **BundleReachable** – Boolean indicating whether there was a problem accessing Microsoft AutoUpdate application bundle.
- **Channel** - The preference for audience
- **Codesigned** – Boolean indicating whether the Update Assistant were codesigned correctly.
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **Exists** – Boolean indicating whether the Update Assistant exists on disk.
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an indication whether Daemon component exists at expected location and whether it is codesigned. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installstatus.helper

This event logs the status of the Microsoft AutoUpdate helper tool. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version

- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an indication whether PrivilegedHelperTool component exists at expected location and whether it is codesigned.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installupdatestask.applaunched

This event indicates Microsoft Autoupdate has detected application launch for a blocked update but could not find matching installer. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains name of the launched application.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)

- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installupdatestask.applaunchwithpendingupdate

This event indicates Microsoft Autoupdate detected application launch for an application with a pending update. Launched application will be terminated. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installupdatestask.codesignverificationfail

This event denotes that CodeSign verification failed for an app update. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device

- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity, name of the updated application and failure code.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installupdatestask.codesignverificationstart

This event denotes that CodeSign verification started for an app update. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity and name of the updated application.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installupdatestask.codesignverificationsuccess

This event denotes CodeSign verification success for an app update. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity and name of the updated application.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installupdatestask.failsilentinstall

This event logs failures while applying silent updates and whether this was a cloned or regular install. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place

- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity and the type of update.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installupdatestask.multiplerelocatablepackage

This event indicates Microsoft Autoupdate has found multiple instances of application entry for a given update package in downloaded manifest. We use this event to ensure the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity and name of update
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installupdatestask.removeclone

This event denotes that a clone was removed. We remove a clone when either the Install On clone process completed, or when a new process starts, and an older cloned version is found in the machine. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version

- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity, name of update, update package name, remove clone status / error details.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installupdatestask.retryfail

This event denotes that errors were encountered during the installation retry process. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity, name of update and whether install should be performed via Install On Clone
- **PipelineInfo_ClientCountry** - The device country (based on IP address)

- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installupdatestask.retryproxyerror

This event logs intra-process communication errors (communication with MAU helper tool). We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity, name of update and details on proxy error reported.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installupdatestask.retryresponse

This event logs that a retry did not work. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device

- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity, name of update, application version, update package name and an indication whether Install On Clone was on, whether install was successful and any errors reported on failure.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installupdatestask.retrysuccess

This event logs a successful update installation after a retry. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity, name of update, application version, update package name and an indication whether Install On Clone was on.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

installupdatestask.setreopengui

This event indicates whether setting preference to reopen GUI after install was successful. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text indicating success of operation. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session
- **Success** – Boolean indicating success of operation.

installupdatestask.updatestatus

This event reports on status of installation task. This event forms part of the update funnel and is used to determine health of application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)

- **DeviceInfo_OsBuild** - The version of the operating system
- **Error** - Indicates any errors encountered during update process, if populated
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **IOC** - Indicates whether Install on Clone feature was used
- **Payload** - Static text to indicate beginning of install process if present
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **Success** - Indicates whether the install process has successfully completed
- **UpdateID** - Identifier for an application update
- **UpdateName** - Name of the update as it appears in the downloaded manifest file
- **UpdatePkg** - Name of the update package being applied

Lifecycle.complimentproclaunch

This event indicates attempt to launch Microsoft Update Assistant from Microsoft AutoUpdate or from Microsoft AutoUpdate from Microsoft Update Assistant. This event is used to determine and ensure health of Microsoft AutoUpdate and Microsoft Update Assistant.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Error** - Any error reported during launch attempt
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting

- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **Reason** - Reason for attempting to launch compliment process
- **SessionId** - The identifier for the session
- **Success** - Indication whether launch attempt was successful

Lifecycle.launch

This event indicates start of Microsoft AutoUpdate or Microsoft Update Assistant. This event is also used to report any issues found during the launch process, as well as reporting method used to launch in the case of Microsoft Update Assistant.

[This event replaces the fba.launch and appdelegate.launch events.]

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Error** - Any error found on launch
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **LaunchedBy** - Method used to launch Microsoft Update Assistant, if applicable
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

Lifecycle.periodiccheck

This event reports on status of MicrosoftAutoUpdate process periodically. Specifically, it reports on what remaining tasks process is waiting on for completion for Update Assistant, and in the case of UI it reports on whether process is terminating due to user inaction. We use this event to determine what is preventing Update Assistant from completing updates and terminating and, whether the UI is terminating due to user inaction.

The following fields are collected:

- **App** - The application process sending the event

- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **dataCollectionDialog** - Boolean indicating whether the process is waiting for user response on Data Collection Dialog
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **forcedUpdateDialog** - Boolean indicating whether the process is waiting for user response on Forced Update Dialog
- **HowToCheck** - How To Check setting
- **isBusy** - Boolean indicating whether the process is busy with active update
- **isInactive** - Boolean indicating whether the process has been waiting for user action for prolonged period of time
- **isWaiting** - Boolean indicating whether the process is waiting for user response on notification
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session
- **SessionLength** - Length of current process session in seconds

Lifecycle.terminate

This event indicates termination of Microsoft AutoUpdate or Microsoft Update Assistant. This event is used to determine the health of Microsoft AutoUpdate and Microsoft Update Assistant.

[This event replaces the fba.terminate and appdelegate.terminate events.]

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier

- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **SessionLength** - Length of current process session in seconds

msupdate.cli.eventhandler

This event is used to calculate the usage of various types of Microsoft Auto Update (MAU) Command Line Interface API.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - The identifier of application that sends command-line interface API to MAU.
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry got received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **EventType** - The type of event that is sent by application to MAU's command-line interface API.
- **HowTocheck** - The preference for checking of updates
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

msupdate.cli.eventhandler.applyupdates.appids

[This event has been removed from current builds of Office, but might still appear in older builds.]

This event indicates a CLI (client-line interface) command was issued to apply an update. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains list of application IDs to be updated.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

msupdate.cli.eventhandler.config

[This event has been removed from current builds of Office, but might still appear in older builds.]

This event indicates Microsoft Autoupdate Command Line Interface module received an Apple event to configure. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)

- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

msupdate.cli.eventhandler.updates

[This event has been removed from current builds of Office, but might still appear in older builds.]

This event indicates Microsoft Autoupdate Command Line Interface module received an Apple event to list updates. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

msupdate.monitor.progress.downloaded

This event indicates that updates were downloaded. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains lists of updates downloaded
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

msupdate.monitor.progress.failure

This event logs a list of queued updates that failed to be applied. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates

- **Payload** - Contains lists of updates.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

msupdate.monitor.progress.finished

This event logs a list of queued updates that completed install. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains lists of updates.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

msupdate.monitor.progress.queued

This event logs a list of queued updates. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier

- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains lists of updates.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

Optinnotificationaction

[This event has been removed from current builds of Office, but might still appear in older builds.]

This event logs the user's response to opt-in dialog for enrolling into silent updates. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains static text representing user selection for opting into Automatic Download and Install.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

sauforcedupdate.autodismiss

This event indicates displayed force update dialog is being dismissed due to user inactivity. This event is used to determine volume of forced updates that proceed without users providing any input to the displayed notification. This event is used to enhance user interface to minimize interruption.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** -The first three octets of the IP address
- **Reason** - Static text
- **SessionId** - The identifier for the session

sauforcedupdate.close

This event indicates user has chosen to close forced update dialog. This event is used to determine volume of forced updates that are postponed by user action. This event is used to enhance user interface to minimize interruption.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device

- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

sauforcedupdate.completeautodismiss

This event indicates that displayed forced update dialog from deadline feature is being dismissed due to user inactivity. This event is used to determine volume of forced updates that proceed without users providing any input to the displayed notification. This event is used to enhance user interface to minimize interruption for deadline feature.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

sauforcedupdate.completeclose

This event indicates successful completion of a forced update. This event is used to help determine health of the

forced update feature.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

sauforcedupdate.display

This event indicates that a forced update dialog has been displayed. This event forms part of a forced update funnel and is used to determine the health of the forced update feature.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged

- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

sauforcedupdate.displayfinalhour

This event indicates that a forced update final hour dialog has been displayed. This event forms part of forced update funnel and is used to determine health of forced update feature.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

sauforcedupdate.done

This event indicates that a forced update has successfully completed. This event forms part of forced update funnel and is used to determine health of forced update feature.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience

- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

sauforcedupdate.enabled

This event is triggered when Microsoft Auto Update (MAU) determines forced update is applicable. This event is used to determine health of forced update feature.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Enabled** - Indicates whether forced update is enabled
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **InvalidUpdates** - Count of forced updates set with invalid update versions
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address

- **SessionId** - The identifier for the session

sauforcedupdate.forcedupdatedismiss

This event indicates that the displayed force update final hour dialog is being dismissed due to user inactivity. This event is used to determine volume of forced updates that proceed without users providing any input to the displayed notification. This event is used to enhance the user interface to minimize interruption.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **Reason** - Static text
- **SessionId** - The identifier for the session

sauforcedupdate.forcequitandupdatenow

This event indicates start of user initiated forced update. This event forms part of funnel and is used to determine health of forced update feature.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier

- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

sauforcedupdate.forceterminate

This event indicates start of the forced update with application being terminated forcefully. This event forms part of the funnel and is used to determine health of forced update feature.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Contains count of applications to be terminated
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

sauforcedupdate.quitandupdatenow

This event indicates that the user has elected to close the application and apply update. This event forms part of a funnel and is used to determine the health of the forced update feature.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

sauforcedupdate.snooze

This event indicates user has elected to postpone forced update. This event forms part of funnel and is used to determine health of forced update feature.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Duration** - Text indicating duration of snooze
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged

- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

sauforcedupdate.terminate

This event indicates start of the forced update with application being terminated. This event forms part of the funnel and is used to determine health of forced update feature.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Contains count of applications to be terminated
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

sauforcedupdate.updatenow

This event indicates user has elected to update application now. This event forms part of funnel and is used to determine health of forced update feature.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version

- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

sauupdateinfoprovider

This event logs whenever a manifest key is missing from a collateral. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains the string used to look for update location or size.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address

- **SessionId** - The identifier for the session

update.applaunchdetected

This event indicates an application was launched while an update was in progress. This event is used to determine the volume of applications that are launched during update and is used for enhancing user experience in future releases.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **Success** - Indicates whether launched app has successfully terminated
- **UpdateID** - Identifier for an application update

update.appterminationreceived

This event indicates that an application with blocked update has been terminated and whether Microsoft Auto Update (MAU) can continue with the update. This event forms part of a funnel and is used to determine the health of application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience

- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Error** - Indicates whether there are other instances of application still running, preventing MAU from continuing
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text to indicate MAU is continuing with update
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** - Identifier for an application update

update.blockedappclosed

This event indicates Microsoft Auto Update (MAU) has detected an application with blocked update has closed and can continue with update. This event forms part of funnel and is used to determine health of application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received.
- **EventInfo_Name** - The name of the telemetry event being logged.
- **EventInfo_Time** - The time at which the logged event took place.
- **HowToCheck** - How to check setting

- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** - Identifier for an application update

update.blockedinstallskip

This event logs an error encountered when trying to skip an application update. This event is critical and is used to investigate reported errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Contains information on error encountered
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

update.clientsession

This event is reported when the status of client device changes, causing Microsoft Update Assistant to pause or resume update process. This event forms part of funnel and is used to determine health of application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience

- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Indicates whether Microsoft Auto Update (MAU) is resuming or pausing
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

update.clonedisablereason

This event logs a condition that an Install-On-Clone feature is disabled for a particular update. We use this event to monitor the health of Install-On-Clone feature and to provide improved service.

The following fields are collected:

- **App** – The application process sending the event
- **AppInfo_Language** – The language the application is running under
- **AppVersionLong** – The application version
- **Channel** – The preference for audience
- **Device_NetworkCountry** – The device county (based on IP address)
- **DeviceID** – The device identifier
- **DeviceInfo_Model** – The hardware model of the device
- **DeviceInfo_NetworkType** – The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** – The version of the operating system
- **Event_ReceivedTime** – The time at which telemetry got received
- **EventInfo_Name** – The name of the telemetry event being logged
- **EventInfo_Time** – The time at which the logged event took place
- **HowTocheck** – The preference for checking of updates
- **PipelineInfo_ClientCountry** – The device country (based on IP address)
- **PipelineInfo_ClientIp** – The first 3 octets of the IP address
- **Reason** – Reason why Install On Clone is disabled for this update.
- **SessionId** – The identifier for the session

update.download.begin

This event indicates start of the application update process. This event forms part of the update funnel and is used to determine the health of application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **IsRepair** - Indicates whether the update is to repair failed update
- **Payload** - Indicates whether a download was attempted previously
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateName** - Name of the update as it appears in the downloaded manifest file

update.download.finish

This event indicates completion of the download phase for application update. This event forms part of the update funnel and is used to determine the health of application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier

- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **IsRepair** - Indicates whether the update is to repair failed update
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** - Identifier for an application update
- **UpdateName** - Name of the update as it appears in the downloaded manifest file

update.downloadresume

This event reports an error encountered while attempting to resume a paused download task. This event is critical and is used to investigate on reported errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Error** - Indicates nature of error encountered
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address

- **SessionId** - The identifier for the session
- **UpdateID** - Identifier for an application update

update.error

This event reports an error encountered while attempting to update registered application. This event is critical and is used to investigate on reported errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Error** - Contains information on nature of error encountered
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Contains information on nature of error encountered
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

update.installcleanupforapp

This event indicates update install has completed and Microsoft Auto Update (MAU) is cleaning up. This event forms part of update funnel and is used to determine health of application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppState** - State of registered application. May indicate error, pending repair, etc.
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)

- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** - Identifier for an application update

update.installupdateforapp

This event is used to report on start of application update install process. This event forms part of update funnel and is used to determine health of application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Error** - Encountered errors if any
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

- **UpdateID** - Identifier for an application update
- **UpdateName** - Name of the update as it appears in the downloaded manifest file

update.installupdateforapp.success

This event reports on the status of install task. This event forms part of update funnel and is used to determine health of application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **ForcedUpdate** - String indication whether an update is forced by IT Admin
- **HowToCheck** - How to check setting
- **Payload** - Indicates whether progress view has been displayed during install process
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **Success** - Success indication returned from install task
- **UpdateID** - Identifier for an application update

Update.InstallVariance

This event is used to calculate critical update health metric for MAU. This event allows us to determine the success metrics of install priority feature and verify the integrity of the feature.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience

- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains the list of Application IDs and their corresponding install priority represented in numbers.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

update.multipleappupdates

This event indicates multiple application updates are in progress in background. This event forms part of update funnel and is used to determine health of application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Contains information on number of applications being updated
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address

- **SessionId** - The identifier for the session

update.previousidnil

This event indicates a repair update package is being downloaded but there is no previous download information. This event is critical and is used to investigate reported errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Error** - Indicates nature of error encountered
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

update.queueinstallforapp

This event indicates that a downloaded update package has been placed in a queue for install. This event forms part of update funnel and is used to determine health of application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device

- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text to indicate application needs to close, if present
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **UpdateID** - Identifier for an application update
- **UpdateName** - Name of the update as it appears in the downloaded manifest file

update.relaunchafterupdate

This event indicates application update has completed and is being re-launched. This event forms part of update funnel and is used to determine health of application updates.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Error** - Contains information on any errors encountered while attempting to relaunch application
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

- **UpdateID** - Identifier for an application update

update.timerforappterminination

This event indicates start/end of timer for checking on the status application. This event comes in a pair and is used to determine all timer objects are removed when application update progresses.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Indicates whether the timer was added or removed
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

updatecore.appregistration

This event logs attempts to register an app and the result/reason. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System

- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains an identifier used to track an update activity, indication whether preference is available, indication if this is re-registration and an indication whether registration is required.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatecore.loadinglaunchagent

This event indicates Launch Agent is being loaded. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatecore.runnstaskcommand

This event reports on an error while attempting to launch a task. This event is critical and is used to investigate reported errors.

The following fields are collected:

- **App** - The application process sending the event

- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Contains path to the command being executed
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

updatecore.server.connectionfail

This event logs errors hit while reaching out to the CDN. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains information on server name, whether server is valid and if server is reachable.

- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatecore.server.nullurl

This event reports an error indicating that a given server could not be reached. This event is used to determine update failure rate caused by network issue.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

updatefilterhelper.cannotretrievebuilddate

We can filter updates via MAU Service only when the update being offered is not older than certain number of days. Here we log that we could not retrieve the date from the app metadata. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier

- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains application ID.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefilterhelper.invalidappid

This event reports an error indicating that no matching manifest files could be found with application id retrieved from web response. This event is used to investigate reported error.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Contains application ID in the web response
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

updatefilterhelper.invalidappidfromwebservices

This event reports an error indicating that application ID retrieved from web response is not in expected format. This event is used to investigate reported error.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Static text
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

updatefilterhelper.invalidresponsefromupdatefiltering

We can filter updates via MAU Service only when the update being offered is not older than certain number of days. Here we log the date is missing from the app metadata. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged

- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains application ID.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefilterhelper.missingbuilddate

We can filter updates via MAU Service only when the update being offered is not older than certain number of days. Here we log the date is missing from the app metadata. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains application ID.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefilterhelper.updatebypassedoldage

We can filter updates via MAU Service only when the update being offered is not older than certain number of days. Here we log the service is bypassed due to old update date. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under

- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains application ID.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.check.error

This event reports an error encountered while checking for updates. This event is critical and is used to investigate reported error.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Code** - Error code
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Domain** - Error domain
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting

- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.check.start

This event logs whenever we initiate a check for updates operation. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains information on updates to offer, registered applications and temporary location downloaded files will save to.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.check.status

This event aggregates the status of the check for update operations (the funnel from searching until downloading). We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)

- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains information on updates to offer, registered applications and temporary location downloaded files will save to.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.check.updatefound

We log whenever a check for updates results in updates found. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.check.updatenotfound

We log whenever a check for updates results in no updates offered due to no updates being found. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.check.uptodate

We log whenever a check for updates results in no updates offered due to all apps being already up to date. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received

- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.offerupdates.invalidappid

This event reports an error while trying to assess whether an update is applicable. This event is critical and is used to investigate reported error.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **CatalogID** - Identifier for the accessed catalog
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **IsNullID** - Indicates whether ID is null
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.offerupdates.minoscheckfail

We log whenever we blocked an update due to not meeting OS requirements. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under

- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains minimum required OS version as specified in downloaded manifest file.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.offerupdates.missingtrigger

This event reports an error while attempting to evaluate triggers from downloaded application update manifest. This critical and is used to investigate reported error.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address

- **SessionId** - The identifier for the session
- **TriggerKey** - Trigger key found in manifest
- **Triggers** - Dictionary of triggers found in manifest

updatefinder.offerupdates.nullbundleforappid

This event indicates Microsoft Autoupdate was unable to load bundle information for the application ID specified in downloaded manifest file. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains application ID.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.offerupdates.updaterulematched

This event denotes that an update was found for an app and baseline. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier

- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains application ID and bundle version information.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.registeredapps

We log the apps that are installed/registered/Controlled by MAU. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains application ID and bundle version information.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.suite.invalidsuiteversion

This event reports on a suite version error while assessing whether an update is applicable. This event is critical and is used to investigate reported error.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **Suite** - Name of suite under consideration

updatefinder.suite.keyvaluemissing

This event reports an error while attempting to add an application to suite. This event is critical and is used to investigate reported error.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged

- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session
- **Suite** - Name of suite application is to be added

updatefinder.suite.missingcollateral

Suite Update - We log whenever a suite update is not applicable due to collateral missing. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Text that indicates the nature of the event.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.suite.staleversion

Suite Update - We log whenever a suite update is not applicable due to baseline version being too old. We log the baseline version and the Suite AppId. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version

- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains suite name.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.suite.updateapplicable

Suite Update - We log whenever a suite update is applicable. We log the baseline version and the Suite AppId. We log the baseline version and the Suite AppId. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains Name, Baseline and update version for the suite.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)

- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.suite.updatenotapplicabledefaultpath

Suite Update - We log whenever a suite update is not offered due not all suite apps being install under the default path. We log the baseline version and the Suite AppId. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains Name, Baseline and update version for the suite.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.suite.updatenotapplicableversion

Suite Update - We log whenever a suite update is not offered due not all suite apps being in the same baseline version. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device

- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains Name, Baseline and update version for the suite.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.suite.updatenotoffered

Suite Update - We log whenever a suite update is not offered due to suite size being larger than individual updates. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains suite name.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatefinder.suite.updateoffered

Suite Update - We log whenever a suite update is offered. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains Name, Baseline and update version for the suite.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatemanager.checkupdate

This event logs number of updates found by Microsoft Autoupdate while checking for available updates. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates

- **Payload** - Contains count of available updates found.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

updatemanager.network

This event logs network availability. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** – The application process sending the event
- **AppInfo_Language** – The language the application is running under
- **AppVersionLong** – The application Version
- **Channel** – The preference for audience
- **Device_NetworkCountry** – The device county (based on IP address)
- **DeviceID** – The device identifier
- **DeviceInfo_Model** – The Hardware Model of the device
- **DeviceInfo_NetworkType** – The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** – The Version of the Operating System
- **Event_ReceivedTime** – The time at which telemetry got received
- **EventInfo_Name** – The name of the telemetry event being logged
- **EventInfo_Time** – The time at which the logged event took place
- **HowTocheck** – The preference for checking of updates
- **PipelineInfo_ClientCountry** – The device country (based on IP address)
- **PipelineInfo_ClientIp** – The first 3 octets of the IP address
- **SessionId** – The identifier for the session
- **ServerReacheable** – Boolean indicating whether the network is available.

updatemanager.updatespending

This event denotes that updates were found and are pending install. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier

- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains indication whether the task is running on main thread, and number of pending updates.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

UpdateStatus.Codesign

This event reports the status from codesign verification Microsoft Update Assistant runs after installing client application updates. We use this event to ensure that we provide packages that are valid and will update the installed application to newest version.

The following fields are collected:

- **App** - The application process sending the event
- **AppID** - Identifier for the application being updated
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Error** - Any errors that were seen during codesign verification process
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

- **Success** - Indicates whether codesign verification was successful
- **UpdateID** - Uniquely identifies applied update
- **UpdateName** - Name of the update as described in update manifest
- **UpdatePkg** - Name of the update package applied

urlutilities.getmauinfo

This event reports an error encountered while accessing Microsoft Auto Update (MAU) application bundle. This event is critical and is used to investigate reported error.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Contains information on error encountered
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

webservices.checkforsilentupdates

This event denotes that silent-update candidates were found. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier

- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains count of updates found and application ID.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

webservices.deltaupdater

This event logs interactions between the client code and the feature gate that controls whether the client should allow for Delta updates. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains response from web-services and updater type to applied.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

webservices.serviceaction

We log any errors resulting from an unexpected webservice response. We use this event for ensuring the

update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains details of action being pushed from web-services.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

webservices.serviceresponse

This event logs requests to MAU Service, response times and Errors. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged

- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains request ID, application name, response time and/or status code.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

webservices.silentupdate

We log requests to check for any "force update" applicability rules, i.e. we must take a user from build N to build N+1 due to some major issue. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains request ID, application name, response time and/or status code.
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address
- **SessionId** - The identifier for the session

webservices.updatefiltering

This event indicates filtering performed on the list of applicable updates via web-services. We use this event to ensure application blocks are working correctly if we have to block an update.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application version

- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The hardware model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** - The version of the operating system
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowToCheck** - How to check setting
- **Payload** - Contains information on number of updates blocked via web-services
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first three octets of the IP address
- **SessionId** - The identifier for the session

webservices.webcontent

We log requests and responses received into webservices. We use this event for ensuring the update process works as expected and to help troubleshoot errors.

The following fields are collected:

- **App** - The application process sending the event
- **AppInfo_Language** - The language the application is running under
- **AppVersionLong** - The application Version
- **Channel** - The preference for audience
- **Device_NetworkCountry** - The device country (based on IP address)
- **DeviceID** - The device identifier
- **DeviceInfo_Model** - The Hardware Model of the device
- **DeviceInfo_NetworkType** - The type of network (Wi-Fi, Wired, Unknown)
- **DeviceInfo_OsBuild** - The Version of the Operating System
- **Event_ReceivedTime** - The time at which telemetry was received
- **EventInfo_Name** - The name of the telemetry event being logged
- **EventInfo_Time** - The time at which the logged event took place
- **HowTocheck** - The preference for checking of updates
- **Payload** - Contains web-service caller ID
- **PipelineInfo_ClientCountry** - The device country (based on IP address)
- **PipelineInfo_ClientIp** - The first 3 octets of the IP address

- **SessionId** - The identifier for the session

webservices.whatsnew

This event is triggered when Microsoft Auto Update (MAU) queries web-services on the "what's new" feature for registered applications. This event is used to determine health of the "what's new" feature.

The following fields are collected:

- **App** -The application process sending the event
- **AppInfo_Language** -The language the application is running under
- **AppVersionLong** -The application version
- **Channel** -The preference for audience
- **Device_NetworkCountry** -The device country (based on IP address)
- **DeviceID** -The device identifier
- **DeviceInfo_Model** -The hardware model of the device
- **DeviceInfo_NetworkType** -The type of network (Wi-Fi, wired, unknown)
- **DeviceInfo_OsBuild** -The version of the operating system
- **Event_ReceivedTime** -The time at which telemetry was received
- **EventInfo_Name** -The name of the telemetry event being logged
- **EventInfo_Time** -The time at which the logged event took place
- **HowToCheck** -How to check setting
- **Payload** -Contains information on number of applications with what's new information
- **PipelineInfo_ClientCountry** -The device country (based on IP address)
- **PipelineInfo_ClientIp** -The first three octets of the IP address
- **SessionId** -The identifier for the session

OneNote sync events

Office.OneNote.Storage.NotebookSyncResult

This event logs notebook sync result. It is used for figuring out how many unique sync targets when calculating OneNote sync score.

The following fields are collected

- **CachedError_Code** - a numbered or alphanumeric code used to determine the nature of the cached error, and/or why it occurred
- **CachedError_Description** – a description of the cached error
- **CachedError_Tag** – indicate where in the code throws the cached error
- **CachedError_Type** – the type of the cached error, e.g. Win32Error, etc.
- **ExecutionTime** – time in milliseconds taken to replicate the notebook
- **Gosid** – global object space ID
- **IdentityType** – identity type, e.g. Windows Live, Org ID, etc.

- **InitialReplicationInSession** – is this replication the first notebook replication after open or not
- **IsBackgroundSync** – is this a background sync or not
- **IsCachedErrorSuppressed** – is the cached error suppressed or not
- **IsCachedErrorUnexpected** – is the cached error unexpected or not
- **IsNotebookErrorSuppressed** – is the notebook level sync error suppressed or not
- **IsNotebookErrorUnexpected** – is the notebook level sync error unexpected or not
- **IsSectionErrorSuppressed** – is the section sync error suppressed or not
- **IsSectionErrorUnexpected** – is the section sync error unexpected or not
- **IsUsingRealtimeSync** – is the notebook sync using modern page content sync or not
- **LastAttemptedSync** – timestamp when the notebook was attempted to be synced last time
- **LastBackgroundSync** – timestamp when the latest background sync was attempted
- **LastNotebookViewedDate** – the date when the notebook was last viewed
- **LastSuccessfulSync** – timestamp when the notebook successfully synced before
- **NeedToRestartBecauseOfInconsistencies** – does the sync need to restart because of inconsistencies or not
- **NotebookErrorCode** – notebook level sync error code saved on notebook graph space
- **NotebookId** – notebook ID
- **NotebookType** – notebook type
- **ReplicatingAgainBecauseOfInconsistencies** – does the sync restart because of inconsistencies or not
- **SectionError_Code** – a numbered or alphanumeric code used to determine the nature of the section sync error, and/or why it occurred
- **SectionError_Description** – a description of the section sync error
- **SectionError_Tag** – indicate where in the code throws the section sync error
- **SectionError_Type** – the type of the section sync error, e.g. Win32Error, etc.
- **Success** – is the notebook sync successful or not
- **SyncDestinationType** – sync destination type, i.e. OneDrive or SharePoint Online
- **SyncId** – a number unique to each notebook sync
- **SyncWasFirstInSession** – is this sync the first sync in current session
- **SyncWasUserInitiated** – is this sync user initiated or not
- **TenantId** – SharePoint tenant ID
- **TimeSinceLastAttemptedSync** – time since last notebook sync attempt
- **TimeSinceLastSuccessfulSync** – time since last successful notebook sync

Office.OneNote.Storage.RealTime.WebSocketSessionInfo

This event logs WebSocket sync result for both OneNote modern page content sync modern hierarchy sync. It is

used for figuring out how many unique sync targets when calculating OneNote sync score. It is also used for OneNote modern sync performance dashboard.

The following fields are collected:

- **CloseReason** - WebSocket close reason, e.g. Abnormal close, etc.
- **DataIsFreshCount** - number of successful pull requests in the WebSocket session
- **DeviceSessionId** - Device session ID
- **DownloadCount** - number of downloads in the WebSocket session
- **Error** - is basically Exception_Type + Exception_Description + Exception_Code + Exception_Tag
- **Exception_Code** - a numbered or alphanumeric code used to determine the nature of an error, and/or why it occurred
- **Exception_Description** - a description of the error
- **Exception_Tag** - indicate where in the code throws the error
- **Exception_Type** - the type of the error, e.g. Win32Error, etc.
- **FirstUpdateSize** - first update message length
- **HasError** - whether there is an error during the WebSocket session
- **IsEducationNotebook** - Is the current notebook education notebook or not
- **IsHierarchyResource** - Is the current resource a page or a section
- **NotebookId** - OneNote notebook ID
- **OperationWithError** - in which operation did the error happen, e.g. WebSocket.Close, WebSocket.Open, etc.
- **ResourceId** - OneNote page or section resource ID
- **SectionId** - OneNote section ID
- **ServerSessionId** - session ID used to correlate WebSocket request to onenote.com
- **SessionDurationInMs** - the duration in milliseconds of the WebSocket session
- **TenantId** - SharePoint tenant ID
- **TimeToFirstUpdateInMs** - time in milliseconds taken to receive first update from the server side after the WebSocket session is established
- **UploadAckCount** - number of acknowledges for upload in the WebSocket session
- **WebUrl** - PII scrubbed web URL

Office.OneNote.Storage.SectionSyncResult

This event logs section sync result. It is used for figuring out how many unique sync targets when calculating OneNote sync score. It is also used for OneNote modern sync performance dashboard.

The following fields are collected

- **Error_Code** - a numbered or alphanumeric code used to determine the nature of an error, and/or why it occurred
- **Error_Description** - a description of the error

- **Error_Tag** - indicate where in the code throws the error
- **Error_Type** - the type of the error, e.g. Win32Error, etc.
- **ErrorLast** - the error code of last seen error
- **ExecutionTime** - time in milliseconds taken to replicate the section
- **InitialReplicationInSession** - is this replication the first notebook replication after open or not
- **IsAttachedViaShortcut** - is the section attached via shortcut or not
- **IsBackgroundSync** - is this a background sync or not
- **IsEncrypted** - is the section encrypted or not
- **IsErrorSuppressed** - is this error suppressed or not
- **IsErrorTransient** - is this error transient or not
- **IsErrorUnexpected** - is this error unexpected or not
- **IsUsingRealtimeSync** - is the section sync using modern page content sync or not
- **NotebookId** - notebook ID
- **NotebookPath** - PII scrubbed notebook URL
- **SectionPath** - PII scrubbed section URL
- **SectionReplicatingIsOutbound** - is this replication an outbound replication or not
- **SectionReplicatingIsSameIdentity** - is this replication based on the same file identity or not
- **SectionResourceId** - OneNote section resource ID
- **Success** - is the section sync successful or not
- **SyncDestinationType** - sync destination type, i.e. OneDrive or SharePoint Online
- **SyncId** - a number unique to each section sync
- **SyncWasFirstInSession** - is this sync the first sync in current session
- **SyncWasUserInitiated** - is this sync user initiated or not
- **TenantId** - SharePoint tenant ID
- **UnmappedGosid** - section ID before applying the mapping GUID

Office.OneNote.Storage.SyncScore

This event logs all negative factors in sync experience that are visible to users. It is used to calculate OneNote sync score, which is a critical metric to evaluate OneNote users' sync experience.

The following fields are collected

- **AutoShowSyncStatus** - whether sync status is auto shown or not
- **Cause** - what caused OneNote pages/sections moved to misplaced sections
- **Context** - an enum categorizes what user is trying to do, e.g. rename a section, reopen a notebook, etc.
- **Error_Code** - a numbered or alphanumeric code used to determine the nature of an error, and/or why it occurred

- **Error_Description** - a description of the error
- **Error_Tag** - indicate where in the code throws the error
- **Error_Type** - the type of the error, e.g. Win32Error, etc.
- **ErrorText** - error text shown in the UI
- **Explanation** - explains what kind of pending outbound changes that need to be moved to misplaced sections
- **fishbowlType** - type of fishbowl, e.g. page fishbowl, section fishbowl, etc.
- **IDS** - an integer identifier for the text shown in the UI
- **idsFishbowl** - an integer identifier for the fishbowl error shown in the UI
- **IsUsingRealtimeHierarchySync** - Is using modern hierarchy sync or not
- **NotebookId** - notebook ID
- **PageSyncUIState** - page sync status string, e.g., UpToDate, Syncing, SaveOffline, SyncError, etc.
- **ServerGosid** - resource ID for newly created conflict page
- **Source** - an enum indicates which event triggered the UI, i.e. created a new redx image, sync error in the sync UI, error dialog displayed, etc.

OneNote.App.Provisioning.MoveLocalNotebookToOnlineNotebookFailed

This event is logged when move of local notebook to drive fails. This scenario is specific to delayed sign-in user. When the user signs in, their local notebook is transferred to their OneDrive storage.

The following fields are collected:

- **ErrorMsg** - The error message corresponding to the failure.

OneNote.Storage.ConnectivityChanged

The event logs if a user has internet connectivity or not. This is used to correlate the other sync health performance metrics by allowing us to ignore events that occur while a user does not have internet connectivity as we do not expect our service latency to be acceptable without internet connectivity. This allows us to calculate an accurate session count for our metrics across slices of customers (per-tenant, per-sector). We also use it to filter error reports as there are numerous sync errors that we expect to occur without network connectivity but that warrant investigation otherwise.

If we do not receive this data, we will not be able to accurately monitor our products performance or determine if errors experienced by a user are expected or require further investigation.

The following fields are collected:

- **InternetConnectivityNowAvailable** - If the connectivity state has been changed so it is now Internet

OneNote.Storage.LegacyInboundLatency

The critical signal used to track the performance of inbound sync operations that communicate directly with SharePoint including correlating information allowing us to monitor and investigate the performance of uploading data to our service. This signal is only collected for the worst performing download in the last 300 seconds (the number of seconds is configurable by Microsoft depending on service performance and condition).

This is used to ensure service health by allowing us to see which tenants are experiencing an unacceptably slow inbound of data to our service, information about the data they are uploading when they experienced the slow inbound and how widespread within a tenant that latency issue is. It is also used to report service health and performance across our customers to measure trends over time and alert on issues automatically for

engineering mitigation. If we do not have this data it will prevent us from ensuring adequate download performance when a user syncs changes from SharePoint to their computer.

The following fields are collected:

- **IsEducationNotebook** - A bool indicating if the notebook is an education notebook
- **NotebookId** - The ID of the notebook that this upload is part of
- **TimeToConfirmSyncedWithServerInMs** - The time in milliseconds it took to perform the upload

OneNote.Storage.LegacyOutboundLatency

The critical signal used to track the performance of outbound sync operations that communicate directly with SharePoint including correlating information allowing us to monitor and investigate the performance of uploading data to our service. This signal is only collected for the worst performing download in the last 300 seconds (the number of seconds is configurable by Microsoft depending on service performance and condition).

This is used to ensure service health by allowing us to see which tenants are experiencing an unacceptably slow outbound of data to our service, information about the data they were uploading when they experienced the slow outbound and how widespread within a tenant that latency issue is. It is also used to report service health and performance across our customers to measure trends over time and alert on issues automatically for engineering mitigation. If we do not have this data, it will prevent us from ensuring adequate performance when syncing users changes up to SharePoint.

The following fields are collected:

- **IsEducationNotebook** - A bool indicating if the notebook is an education notebook
- **NotebookId** - The ID of the notebook that this upload is part of
- **TimeToConfirmSyncedWithServerInMs** - The time in milliseconds it took to perform the upload

OneNote.Storage.RealTime.FileDataObjectDownload

The critical signal used to track performance when a user inbounds a file data object (i.e. an embedded file or image) which is downloaded directly from our service and not as part of a sync operation on a page, section or notebook. This signal is only collected for the worst performing download in the last 300 seconds (the number of seconds is configurable by Microsoft depending on service performance and condition).

This is used to ensure service health and performance by allowing us to see which tenants are experiencing an unacceptably slow download of data from our service, and how widespread within a tenant that latency issue is, and report our behavior over time allowing us to measure service performance trends. If we see an unacceptable latency for a file object, we will also use this data to correlate that with other signals from the client and service regarding the object to make improvements to our download process. We also split the data based on the extension of the file object downloaded as we have different expectations based on whether the file is presented inline in our canvas (e.g. an image) or is a non-inline file (such as a text document). If we do not receive this data, it will prevent us from monitoring the performance of these downloads

The following fields are collected:

- **FileSizeInBytes** - The size of the file being downloaded in bytes
- **IsImage** - A bool determining if the file being downloaded has an extension that matches a pre-determined list of common image formats (.bmp, .emf, .gif, .jpe, .jpeg, .jpg, .png) that we display inline in the canvas
- **TimeToDownload** - The length of time it took to successfully download the FDO from our blob storage to the device

OneNote.Storage.RealTime.WebSocketDownload

The critical signal used to track performance of inbound sync operations including correlating information allowing us to monitor and investigate the performance of downloading data from our service (onenote.com). This signal is only collected for the worst performing download in the last 300 seconds (the number of seconds is configurable by Microsoft depending on service performance and condition).

This is used to ensure service health by allowing us to see which tenants are experiencing an unacceptably slow inbound of data from our service, information about the data they were downloading when they experienced the slow inbound and how widespread within a tenant that latency issue is. It is also used to report service health and performance across our customers to measure trends over time and alert on issues automatically for engineering mitigation.

If we see an unacceptable latency for a section or notebook we will also use this data to correlate that with other signals from the client and service regarding the same document to identify client-side performance regressions allowing us to deliver a more performant service.

If we do not receive this data, we will be unable to monitor the performance of this aspect of our service, or the impact of server side changes we may find necessary due to usage or other factors.

The following fields are collected:

- **DeviceSessionId** - The ID of the device session
- **IsEducationNotebook** - A bool indicating if the notebook is an education notebook
- **IsHierarchyResource** - A bool indicating if the resource is a hierarchy resource
- **NotebookId** - The ID of the notebook that this upload is part of
- **ResourceId** - The ID of the resource that we are uploading
- **SectionId** - The ID of the section that this upload is part of
- **ServerSessionId** - The ID of the server session that this upload is part of
- **TimeToConfirmSyncedWithServerInMs** - The time in milliseconds between a user navigating to a page and the replication stack confirming that page is in sync with the server.
- **TimeToFirstUpdateInMs** - The time in milliseconds between the sync engine beginning inbound replication of a page and that replication operation reaching the in sync with the server state.

OneNote.Storage.RealTime.WebSocketUpload

The critical signal used to track the performance of outbound sync operations including correlating information allowing us to monitor and investigate the performance of uploading data to our service (onenote.com)

This is used to ensure service health by allowing us to see which tenants are experiencing an unacceptably slow outbound of data to our service, information about the data they were uploading when they experienced the slow outbound and how widespread within a tenant that latency issue is. It is also used to report service health and performance across our customers to measure trends over time and alert on issues automatically for engineering mitigation. We will also use this data to track the impact and effectiveness of improvements we make to our clients and services.

If we see an unacceptable latency for a section or notebook we will also use this data to correlate that with other signals from the client and service regarding the same document to identify performance regressions allowing us to deliver a more performant experience.

If we do not receive this data, we will be unable to monitor the performance of this aspect of our service, or the impact of server side changes we may find necessary due to usage or other factors.

The following fields are collected:

- **DeviceSessionId** - The ID of the device session
- **IsEducationNotebook** - A bool indicating if the notebook is an education notebook
- **IsHierarchyResource** - A bool indicating if the resource is a hierarchy resource
- **IsWorstTime** - A bool indicating if the time is a regular upload event, or the worst time we saw on this client in the last 300 seconds (the number of seconds is configurable by Microsoft depending on service performance and condition).
- **NotebookId** - The ID of the notebook that this upload is part of
- **RecommendedPutIntervalInMs** - The time the service has communicated to the client as its recommended put interval
- **ResourceId** - The ID of the resource that we are uploading
- **SectionId** - The ID of the section that this upload is part of
- **SenderRequestId** - The ID of the sender performing the upload
- **ServerSessionId** - The ID of the server session that this upload is part of
- **UploadNonSuspendedTimeInMs** - The time in milliseconds it took to perform the upload excluding the time when the application was suspended
- **UploadTimeInMs** - The time in milliseconds it took to actually perform the upload
- **WaitTimeInMs** - The time in milliseconds between an upload being requested and an upload starting
- **WebUrl** - The WebUrl of the upload (Logged as a PiiWz)

OneNote.Storage.SyncHealth

The critical signal used to track errors and exceptions that have occurred inside the sync stack in the OneNote client allowing us to monitor and mitigate these unexpected conditions.

This is used to ensure service health by allowing us to see error reports from the clients in near real time, which lets us respond to sync issues as they arise. It is also used to identify how widespread an issue is, and how severe by cross-referencing the error tag with the client code to identify the source of failure. We also aggregate this data to get information on our performance over time and the impact and effectiveness of improvements we make to our clients and services. If we do not have this data, we won't be able to proactively respond to error conditions in our sync service without customer escalation.

The following fields are collected:

- **Service** - The sync service the client was using when the error occurred (Legacy or Modern Sync)
- **Tag** - The tag (an identifying value) representing the error that the client encountered during the sync operation

OneNote.Sync.CreateNotebookFailed

This event is logged when creation of a notebook fails.

The following fields are collected:

- **NetworkConnection** - Logs the connection type that the device is currently on e.g. Wi-Fi, offline, 3G
- **ServerType** - Logs the server type where notebook was to be created.

OneNote.Sync.FirstRunError

This event is logged when syncing of Quick Notes failed for a user during their First Run Experience on a device.

This is specific to the First Run scenario.

The following fields are collected:

- **NetworkConnection** - Logs the connection type that the device is currently on e.g. Wi-Fi, offline, 3G
- **ServerType** - Logs the server type where Quick Notes notebook was to be created

Services Configuration events

No required service data events are collected by Services Configuration.

Telemetry events

app.deep.link

This event helps to track the usage of calendar meeting launch, across different endpoints. This event lets us detect two things when a meeting is launched via Skype for Business, and when a meeting is launched via Teams, and if the Teams app is installed.

The following fields are collected:

- **account** - Hashed account information which performed the action
- **action_type** - action performed, such as launch meeting or install application
- **application** - Application that was launched via a deep link, such as Teams or Skype for Business
- **context** - The experience navigated to within the app, e.g., office_union - word, office_union – excel, etc.
- **source** - The origin of an action, for example, initiated from the user, automatically by the client, etc.

Office.Android.DocsUI.PaywallControl.PaywallOperationMetrics

[This event was previously named Office.Android.DocsUI.Views.PaywallOperationMetrics.]

Microsoft uses this to get the health of the feature, success, or error rates for the user for purchases, to ensure appropriate investments to improve the customers' purchase experience across mobile platforms.

The following fields are collected:

- **OperationTimeInMs** - Time taken for the purchase operation to complete (long – milliseconds)
- **PaywallOperationResult** - Success / Error Code / User Canceled (Enum / int – finite)
- **PaywallOperationType** - Kind of Paywall operation (enum/ int - finite)

Office.Android.DocsUI.PaywallControl.PaywallSessionData

[This event was previously named Office.Android.DocsUI.Views.PaywallSessionData.]

Session-based metadata when Paywall UI is shown to the user. Microsoft uses this to get the user journey, and understand the device and OS versions the user is using, to help make decisions on investments in improving the experience in these areas.

The following fields are collected:

- **App Version** - Version code of the consuming application
- **ClientId** - Anonymous non-PII unique device identifier (guid / string)
- **Entry Point** - Unique identifier for contextual or constant entry points from the consuming application
- **isTablet** - Whether the device is showing tablet UX

- **OSVersion** - Android OS version of the device
- **SessionId** - Guid: Unique Paywall session identifier

Office.FirstRun.Apple.TelemetryOptIn

This event is collected for Office applications running under Apple platforms. The event is used to monitor the health of our telemetry opt-in flow in First Run Experience. We collect a code that denotes what type of diagnostic data collection option was selected by the user.

The following fields are collected:

- **Data_EventId** – A code indicating the diagnostic data collection preference selected by the user.

Office.OneNote.GetSharePointIdsForDocument

The data collected logs the failure and success of fetching the SharePoint (SPO) IDs for a document URL. The success and the failure (including the reason for failure) of the call is logged for all platforms. This marker is required to track and diagnose the health of the call made to get the IDs. The IDs are required to have a OneNote page (belonging to SharePoint-stored notebooks) data displayed in the feed.

The following fields are collected:

- **ErrorCode** - int value of error
- **ErrorMessage** - string describing error
- **FailureType** - string to determine type of error
- **HttpStatusCode** - HTTP error code for network call
- **InnerErrorCode** - int code
- **InnerErrorMesage** - message for error
- **IsSuccess** - Boolean value for is signal succeeded

Office.OneNote.GetSharePointIdsForDocumentW32Old

The telemetry logs the failure scenarios and success of fetching the SharePoint (SPO) IDs for a Document URL. The success and the failure (including the reason for failure) of the call is logged. This is only logged in the old win32 platform. This marker is required to track and diagnose the health of the call made to get the IDs. The IDs are required to have the OneNote page (belonging to SharePoint-stored notebooks) data displayed in the feed.

The following fields are collected:

- **ErrorCode** - int value of error
- **ErrorMessage** - string describing error
- **FailureType** - string to determine type of error
- **HttpStatusCode** - HTTP error code for network call
- **InnerErrorCode** - int code
- **InnerErrorMesage** - message for error
- **IsSuccess** - Boolean value for is signal succeeded

Office.System.GracefulExit.GracefulAppExitDesktop

The event is triggered by a graceful application termination for Office client applications such as, but not limited to, Word, Excel, PowerPoint, and Outlook. We use Graceful Exit to measure the health of Office client products. It is intended to be a business-critical signal used by Office engineers to infer product stability.

The following fields are collected:

- **AppBuild** - Build version identifier for the affected process.
- **AppMajor** - Major version identifier for the affected process.
- **AppMinor** - Minor version identifier for the affected process.
- **AppRevision** - Build version identifier for the affected process.
- **BootCompleted** – Did Office process complete boot.
- **DetectionTime** - The time when the unexpected exit was detected.
- **EcsETag** - An experiment identifier for the process.
- **HasEdit** – Was document editing occurring during the Office process.
- **HasOpen** – Was document opened during the Office process.
- **InstallMethod** - Whether the current build of Office was upgraded from, rolled back to, or a fresh install.
- **OfficeUILang** – Language of the Office process.
- **PreviousBuild** - Previously installed build version.
- **SafeMode** – Was Office process in safe mode.
- **SessionId** - A unique identifier of the process.
- **SessionInitTime** - The time when the affected process started.

Office.System.IdentityChanged

User identity information required to fulfill data subject requests.

The following fields are collected:

- **IdentityChanged** - Always true. The identity changed.
- **TimerDetectedChange** - Whether the change was detected by regularly timed ping.

Office.System.PrivacyFallbackToSettingsStore

Used to determine if there are failures with reading the user's privacy settings from the Roaming store.

The following fields are collected:

- **Tag** - The code tag indicating which setting has fallen back to the settings store.

Office.System.SessionDataO365

Metadata required to isolate a failure reproduction.

The following fields are collected:

- **AppId** - Identifier for what Office application this data refers to.
- **ApplicationArchitecture** - What processor architecture Office is built for.
- **AppVersionBuild** - The Build version of the Office application.
- **AppVersionMajor** - The Major version of the Office application.
- **AppVersionMinor** - The Minor version of the Office application.
- **AppVersionUpdate** - The Build Revision of the Office application.
- **CollectorVersion** - A version identifier for the client collection logic.
- **DeviceHash** - A one-way hash of the operating system device identifier.
- **DeviceName** - Name of the Device Office is run on.
- **Domain** - Domain of the Device Office is run on.

- **IsCeip** - Whether the install of Office was enrolled in the defunct Customer Experience Improvement Program.
- **IsDebug** - Whether this is a debug build of Office.
- **IsImmersive** - Whether the Office application is a Universal Windows or Immersive application.
- **IsLaptop** - Whether the device Office is running on is a laptop.
- **IsMicrosoftInternal** - Whether the Windows user running Office is a Microsoft employee.
- **IsO365** - Whether the Office installation is part of the defunct Outlook 365 program.
- **IsTablet** - Whether the device Office is running on is a tablet.
- **IsTerminalServer** - True/false is terminal server client
- **MaxMemory** - The maximum amount of random-access memory available to the device running Office.
- **OsArchitecture** - The CPU architecture the operating system running Office is built for,
- **OsVersionBuild** - The Build version of the Operating System.
- **OsVersionMajor** - The Major version of the Operating System.
- **OsVersionMinor** - The Minor version of the Operating System.
- **OsVersionUpdate** - OS build revision
- **ProcessFileName** - The running application's executable name.
- **ProcessorArchitecture** - What processor architecture Office is running on.
- **ProcessorFrequency** - The speed of the processor on the devices Office is running on in Megahertz.
- **SessionStart** - The time at which the running Office process started.
- **UserName** - The name of the account running Office.

Office.System.SystemHealthCoreMetadata

Metadata required to isolate a failure reproduction.

The following fields are collected:

- **AppBuild** - The Build version of the Office application.
- **AppBuildRevision** - The Build Revision of the Office application.
- **AppMajorVer** - The Major version of the Office application.
- **AppMinorVer** - The Minor version of the Office application.
- **CID** - Pseudonymized user identity
- **CollectibleClassifications** - The set of data classifications that can be collected.
- **CollectionTime** - Time at which the metadata was collected.
- **DeviceManufacturer** - The manufacturer of the device Office is being run on.
- **DeviceModel** - The model of the device Office is being run on.
- **FirstRunTime** - The first time an Office application was run.
- **IsClickToRunInstall** - Whether the Office application was installed using Click -To-Run

- **IsDebug** - Whether this is a debug build of Office.
- **IsLabMachine** - Whether Office is being run in a Microsoft lab.
- **IsLaptop** - Whether the device Office is running on is a laptop.
- **IsMsftInternal** - Whether the Windows user running Office is a Microsoft employee.
- **IsSubscription** - Whether the Office application is installed under a subscription license.
- **IsTablet** - Whether the device Office is running on is a tablet.
- **IsTerminalServer** - Whether Office is being run on a terminal server.
- **MsoAppId** - Identifier for what Office application this data refers to.
- **OfficeArchitectureText** - What processor architecture Office is built for.
- **OsBuild** - The Build version of the Operating System.
- **OsBuildRevision** - OS build revision
- **OSEnvironment** - Identifier for what environment Office is running on.
- **OsMajorVer** - The Major version of the Operating System.
- **OsMinorVer** - The Minor version of the Operating System.
- **OSVersionString** - The Operating System version as a string.
- **ProcessorArchitecture** - What processor architecture Office is running on.
- **ProcessorCount** - The count of processors on the device Office is running on.
- **ProcSpeedMHz** - The speed of the processor on the devices Office is running on in Megahertz.
- **RamMB** - The amount of RAM available in the device Office is run on.
- **SqmUserId** - A random identifier for the install of Office.

Office.System.SystemHealthDesktopSessionLifecycleAndHeartbeat

Provides information on system health metrics.

The following fields are collected:

- **InstallMethod** - Whether the current build of Office was upgraded from, rolled back to, or a fresh install.
- **OfficeArchitectureText** - The architecture of Office product as string (e.g. x86, arm).
- **PreviousBuild** - The version of Office this build was upgraded to or rolled back from.
- **State** - State which the session changed to.
- **Time** - Time when the session state changed.

Office.System.SystemHealthEssentialIdentityCount

Collects the count of signed-in user identities

The following fields are collected:

- **AllIdentityCount** - Count of all identities
- **ValidIdentityCount** - Count of validated identities

Office.System.SystemHealthEssentialMetadataAllIdentities

Monitors the state of accounts recognized by Office in this session. Used to isolate a failure to an account login type if the failure is specific to a type.

The following fields are collected:

- **CollectionTime** - The time at which the identity information was collected.
- **IdentityType** - The type of authentication or account
- **IdentityUniqueId** - Pseudonymized identity identifier
- **IdentityUniqueIdHashed** - One-way hash of the identity unique ID

Office.System.SystemHealthMetadataApplicationAdditional

Metadata required to isolate a failure reproduction.

The following fields are collected:

- **Alias** - If the user running Office is a Microsoft employee, their company internal alias.
- **AppBuild** - The Build version of the Office application.
- **AppBuildRevision** - The Build Revision of the Office application.
- **AppMajorVer** - The Major version of the Office application.
- **AppMinorVer** - The Minor version of the Office application.
- **CID** - Pseudonymized user identity
- **CollectibleClassifications** - The set of data classifications that can be collected.
- **DeviceManufacturer** - The manufacturer of the device Office is being run on.
- **DeviceModel** - The model of the device Office is being run on.
- **DeviceProcessorModel** - The processor model of the device Office is run on.
- **DigitizerInfo** - Information about the digitizer attached to the device Office is run on.
- **DomainName** - The name of the domain the machine running Office is joined to (if any).
- **FirstRunTime** - The first time an Office application was run.
- **HorizontalResolution** - Horizontal screen resolution
- **IsDebug** - Whether this is a debug build of Office.
- **IsImmersive** - Whether the Office application is a Universal Windows or Immersive application.
- **IsJoinedToDomain** - Whether the device running Office is domain joined.
- **IsLabMachine** - Whether Office is being run in a Microsoft lab.
- **IsLaptop** - Whether the device Office is running on is a laptop.
- **IsMsftInternal** - Whether the Windows user running Office is a Microsoft employee.
- **IsOEMInstalled** - Whether the running Office application was installed by an OEM.
- **IsRunAsAdmin** - Whether the Office application is running as Administrator.
- **IsSubscription** - Whether the Office application is installed under a subscription license.
- **MsoAppId** - Identifier for what Office application this data refers to.

- **NumProcPhysCores** - Number of physical cores in the processor.
- **OfficeBuild** - The Build version of the Office shared libraries.
- **OfficeBuildRevision** - The Build Revision version of the Office shared libraries.
- **OfficeMajorVer** - The Major version of the Office shared libraries.
- **OfficeMinorVer** - The Minor version of the Office shared libraries.
- **OsBuild** - The Build version of the Operating System.
- **OsBuildRevision** - OS build revision
- **OsMajorVer** - The Major version of the Operating System.
- **OsMinorVer** - The Minor version of the Operating System.
- **PowerPlatformRole** - An identifier of the OEM preferred computer role of the device Office is run on.
- **ProcessFileName** - The running application's executable name.
- **ProcessorCount** - The count of processors on the device Office is run on.
- **RamMB** - The amount of RAM available in the device Office is run on.
- **SqmUserId** - A random identifier for the install of Office.
- **StudyId** - Software Quality Metrics study identifier.
- **VerticalResolution** - Vertical screen resolution
- **WinUserActType** - Whether the Windows user running Office is a local administrator, power user, or normal user.

Office.System.SystemHealthMetadataApplicationAndLanguage

Metadata is required to isolate a failure reproduction.

The following fields are collected:

- **AppBuild** - The Build version of the Office application.
- **AppBuildRevision** - The Build Revision of the Office application.
- **AppMajorVer** - The Major version of the Office application.
- **AppMinorVer** - The Minor version of the Office application.
- **AppState** - Identifier for what state the Office application is in.
- **Click2RunPackageVersionBuild** - The Build version of the click-to-run installer package.
- **Click2RunPackageVersionMajor** - The Major version of the click-to-run installer package.
- **Click2RunPackageVersionMinor** - The Minor version of the click-to-run installer package.
- **Click2RunPackageVersionRevision** - The Build Revision of the click-to-run installer package.
- **DistributionChannel** - The channel by which Office was distributed.
- **InstallType** - An identifier for the method by which Office was installed.
- **IsClickToRunInstall** - Whether the Office application was installed using click-to-run
- **IsDebug** - Whether this is a debug build of Office.

- **IsImmersive** - Whether the Office application is a Universal Windows or Immersive application.
- **IsMsftInternal** - Whether the Windows user running Office is a Microsoft employee.
- **IsOEMInstalled** - Whether the running Office application was installed by an OEM.
- **IsRunAsAdmin** - Whether the Office application is running as Administrator.
- **IsSubscription** - Whether the Office application is installed under a subscription license.
- **MsoAppId** - Identifier for what Office application this data refers to.
- **OfficeArchitectureText** - What processor architecture Office is built for.
- **OfficeBuild** - The Build version of the Office shared libraries.
- **OfficeBuildRevision** - The Build Revision version of the Office shared libraries.
- **OfficeMajorVer** - The Major version of the Office shared libraries.
- **OfficeMinorVer** - The Minor version of the Office shared libraries.
- **OfficeMuiCount** - The count of Office language packs installed.
- **OfficeSkuLanguage** - The installed SKU language.
- **OfficeSkuLanguageTag** - The installed SKU language.
- **OfficeUiLang** - The User Interface language for the Office application.
- **OfficeUiLangTag** - The User Interface language for the Office application.
- **ProcessFileName** - The running application's executable name.
- **SqmAppId** - Identifier for what Office application this data refers to.

Office.System.SystemHealthMetadataDelayedLogin

User identity information required to fulfill data subject requests.

The following fields are collected:

- **CID** - Pseudonymized user identity

Office.System.SystemHealthMetadataDevice

Metadata required to isolate a failure reproduction.

The following fields are collected:

- **CollectionTime** - Time at which the metadata was collected.
- **ComputerSystemProductUuidHash** - One-way hash of Motherboard UUID.
- **DeviceClass** - An identifier for the type of device Office is being run on.
- **DeviceMake** - Hardware system family identifier of the device Office is run on.
- **DeviceManufacturer** - The manufacturer of the device Office is run on.
- **DeviceModel** - The model of the device Office is being run on.
- **DigitizerInfo** - Information about the digitizer attached to the device Office is run on.
- **IsLaptop** - Whether the device Office is running on is a laptop.
- **IsTablet** - Whether the device Office is running on is a tablet.

- **LicensingACID** - Licensing identifier for the install of Office.
- **MachineName** - The name of the device Office is being run on.
- **NumProcPhysCores** - Number of physical cores in the processor.
- **NumProcShareSingleCache** - The Number of processors sharing a single cache on the device Office is run on.
- **NumProcShareSingleCore** - The number of processors per physical core on the device Office is run on.
- **OlsLicenseId** - Licensing service identifier for the install of Office.
- **Platform** - An identifier for what environment Office is run on.
- **PowerPlatformRole** - An identifier of the OEM preferred computer role of the device Office is run on.
- **ProcessorCount** - The count of processors on the device Office is run on.
- **ProcSpeedMHz** - The speed of the processor on the device Office is run on in Megahertz.
- **ProcType** - The architecture of the processor.
- **ProcTypeText** - The type of the processor on the device Office is run on.
- **RamMB** - The amount of RAM available in the device Office is run on.
- **SusClientId** - The Windows Update ID of the device Office is run on.
- **SystemFamily** - Hardware system family identifier of the device Office is run on.
- **SystemSKU** - Hardware system SKU identifier of the device Office is run on.
- **SysVolFreeSpaceMB** - The amount of free space available on the System volume in megabytes.
- **SysVolSizeMB** - The amount of space on the System volume in megabytes.
- **WindowsErrorReportingMachineId** - Windows Error Reporting assigned machine identifier of the device Office is run on.
- **WindowsSqmMachineId** - Windows assigned machine identifier of the device Office is run on.

Office.System.SystemHealthMetadataDeviceConsolidated

Metadata required to isolate a failure reproduction.

The following fields are collected:

- **BootDiskType** - Disk or solid-state drive
- **ComputerSystemProductUuidHash** - One-way hash of Motherboard UUID.
- **DeviceClass** - An identifier for the type of device Office is being run on.
- **DeviceManufacturer** - The manufacturer of the device Office is run on.
- **DeviceModel** - The model of the device Office is being run on.
- **DeviceProcessorModel** - The processor model of the device Office is run on.
- **DigitizerInfo** - Information about the digitizer attached to the device Office is run on.
- **HasSpectreFix** - Whether the processor of the device Office is run on has a Spectre fix.
- **IsLaptop** - Whether the device Office is running on is a laptop.

- **IsTablet** - Whether the device Office is running on is a tablet.
- **MachineName** - The name of the device Office is being run on.
- **NumProcPhysCores** - Number of physical cores in the processor.
- **NumProcShareSingleCache** - The Number of processors sharing a single cache on the device Office is run on.
- **NumProcShareSingleCore** - The number of processors per physical core on the device Office is run on.
- **Platform** - An identifier for what environment Office is run on.
- **PowerPlatformRole** - An identifier of the OEM preferred computer role of the device Office is run on.
- **powerPlatformRole** - An identifier of the OEM preferred computer role of the device Office is run on.
- **ProcessorCount** - The count of processors on the device Office is run on.
- **ProcSpeedMHz** - The speed of the processor on the device Office is run on in Megahertz.
- **ProcType** - The architecture of the processor.
- **ProcTypeText** - The type of the processor on the device Office is run on.
- **RamMB** - The amount of RAM available in the device Office is run on.
- **SusClientId** - The Windows Update ID of the device Office is run on.
- **SysVolFreeSpaceMB** - The amount of free space available on the System volume in megabytes.
- **SysVolSizeMB** - The amount of space on the System volume in megabytes.
- **sysVolSizeMB** - The amount of space on the System volume in megabytes.
- **WindowsErrorReportingMachineId** - Windows Error Reporting assigned machine identifier of the device Office is run on.
- **WindowsSqmMachineId** - Windows assigned machine identifier of the device Office is run on.

Office.System.SystemHealthMetadataOperatingSystem

Metadata required to isolate a failure reproduction.

The following fields are collected:

- **CollectionTime** - The time this event was queued for upload
- **IsTerminalServer** - True/false is terminal server client
- **OsBuild** - The Build version of the Operating System.
- **OsBuildRevision** - OS build revision
- **OSEnvironment** - Windows, iOS, Mac, Android, etc.
- **OsMajorVer** - The Major version of the Operating System.
- **OsMinorVer** - The Minor version of the Operating System.
- **OSSDKVersionCode** - Version identifier for the Operating System SDK.
- **OsSku** - OS SKU
- **OsSuite2** - Operating System suite identifier.

- **OSVersionString** - Operating System version identifier.
- **ServicePackMajorVer** - OS service pack major version
- **ServicePackMinorVer** - OS service pack minor version

Office.System.SystemHealthMetadataOperatingSystemDevice

Metadata required to isolate a failure reproduction.

The following fields are collected:

- **CollectionTime** - The time this event was queued for upload
- **DeviceClass** - An identifier for the type of device Office is being run on.
- **DeviceManufacturer** - The manufacturer of the device Office is run on.
- **DeviceModel** - The model of the device Office is being run on.
- **DigitizerInfo** - Information about the digitizer attached to the device Office is run on.
- **IsLaptop** - Whether the device Office is running on is a laptop.
- **IsTablet** - Whether the device Office is running on is a tablet.
- **IsTerminalServer** - True/false is terminal server client
- **MachineName** - The name of the device Office is being run on.
- **NumProcPhysCores** - Number of physical cores in the processor.
- **NumProcShareSingleCache** - The Number of processors sharing a single cache on the device Office is run on.
- **NumProcShareSingleCore** - The number of processors per physical core on the device Office is run on.
- **OsBuild** - The Build version of the Operating System.
- **OsBuildRevision** - OS build revision
- **OSEnvironment** - Windows, iOS, Mac, Android, etc.
- **OsMajorVer** - The Major version of the Operating System.
- **OsMinorVer** - The Minor version of the Operating System.
- **OSSDKVersionCode** - Version identifier for the Operating System SDK.
- **OsSku** - OS SKU
- **OsSuite2** - Operating System suite identifier.
- **OSVersionString** - Operating System version identifier.
- **Platform** - An identifier for what environment Office is run on.
- **PowerPlatformRole** - An identifier of the OEM preferred computer role of the device Office is run on.
- **ProcessorCount** - The count of processors on the device Office is run on.
- **ProcSpeedMHz** - The speed of the processor on the device Office is run on in Megahertz.
- **ProcTypeText** - Processor type

- **RamMB** - The amount of RAM available in the device Office is run on.
- **ServicePackMajorVer** - OS service pack major version
- **ServicePackMinorVer** - OS service pack minor version
- **SysVolFreeSpaceMB** - The amount of free space available on the System volume in megabytes.
- **SysVolSizeMB** - The amount of space on the System volume in megabytes.

Office.System.SystemHealthMetadataOS

Metadata required to isolate a failure reproduction.

The following fields are collected:

- **CountryRegion** - Country/Region identifier Operating System setting.
- **HorizontalResolution** - Horizontal screen resolution
- **IsTerminalServer** - True/false is terminal server client
- **KeyboardLanguage** - Device Keyboard language identifier
- **KeyboardLanguageTag** - Device Keyboard language identifier
- **OfficeWvd** - Identifies what state Windows Virtual Desktop is in.
- **OsBuild** - The Build version of the Operating System.
- **OsBuildRevision** - OS build revision
- **OSEnvironment** - Windows, iOS, Mac, Android, etc.
- **OsLocale** - Operating System locale identifier.
- **OsLocaleTag** - Operating System locale identifier.
- **OsMajorVer** - The Major version of the Operating System.
- **OsMinorVer** - The Minor version of the Operating System.
- **OSSDKVersionCode** - Operating System SDK Version identifier.
- **OsSku** - Operating System SKU identifier.
- **OsSuite2** - Operating System suite identifier.
- **OsUiLang** - Operating System user interface language.
- **OSVersionString** - Operating System version identifier.
- **ScreenDepth** - Screen depth
- **ScreenDpi** - Screen dpi
- **ServicePackMajorVer** - OS service pack major version
- **ServicePackMinorVer** - OS service pack minor version
- **SystemLocale** - Operating System default locale
- **SystemLocaleTag** - Operating System default locale
- **TimeZoneBiasInMinutes** - The difference in minutes between local time and UTC.
- **VerticalResolution** - Vertical screen resolution

Office.System.SystemHealthMetadataScreenCultureUserSqmId

Metadata required to isolate a failure reproduction.

The following fields are collected:

- **Alias** - Microsoft employee or automated user alias
- **CID** - Pseudonymized user identity
- **CollectibleClassifications** - Data classifications that can be collected according to the client privacy settings
- **CollectionTime** - The time this event was queued for upload
- **CountryRegion** - Country/Region identifier Operating System setting.
- **DomainName** - Domain name of Microsoft domain
- **HorizontalResolution** - Horizontal screen resolution
- **IntegratedScreenSize** - Size of the integrated screen.
- **IsJoinedToDomain** - True/false is the client domain joined
- **IsLabMachine** - Is a Microsoft testing lab machine
- **IsMsftInternal** - True/false is the machine in Microsoft corporate domain
- **IsSubscription** - Whether the Office application is installed under a subscription license.
- **KeyboardLanguage** - Device Keyboard language identifier
- **KeyboardLanguageTag** - Device Keyboard language identifier
- **OsLocale** - Operating System locale identifier.
- **OsLocaleTag** - Operating System locale identifier.
- **OsUiLang** - Operating System user interface language.
- **ScreenDepth** - Screen depth
- **ScreenDpi** - Screen dpi
- **ScreenXDpi** - Screen X DPI
- **ScreenYDpi** - Screen Y DPI
- **SqmUserId** - A random identifier for the install of Office.
- **StudyId** - Software Quality Metrics study identifier.
- **SystemLocale** - Operating System default locale
- **SystemLocaleTag** - Operating System default locale
- **TimeZoneBiasInMinutes** - The difference in minutes between local time and UTC.
- **VerticalResolution** - Vertical screen resolution
- **WinUserActType** - Whether the Windows user running Office is a local administrator, power user, or normal user.

Office.System.SystemHealthOfficeLensIdentity

User identity information required to fulfill data subject requests.

The following fields are collected:

- **CID** - Pseudonymized user identity

Office.System.SystemHealthRollbackSessionMetadata

Metadata required to isolate a failure reproduction.

The following fields are collected:

- **InstallMethod** - New install, update, or rollback
- **IsSubscription** - Whether the Office application is installed under a subscription license.
- **PreviousBuild** - Previously installed build version

Office.System.SystemHealthSessionLifecycleAndHeartbeat

Provides information on system health metrics.

The following fields are collected:

- **InstallMethod** - Whether the current Office install was upgraded from, rolled back to, or a fresh install.
- **InteractionSessionID** - Session Identifier.
- **PreviousBuild** - The version of Office this build was upgraded to or rolled back from.
- **State** - State which the session changed to.
- **Time** - Point at which the session state changed.

Office.System.SystemHealthSessionStartTime

Used with crash data to separate early vs late crashes (i.e. determine if the user used the app for some period before the crash)

The following fields are collected:

- **SessionStart** - Time at which telemetry starts processing data.

Office.System.SystemHealthUngracefulAppExitDesktop

The event is triggered by an abnormal application termination (Example: task manager kill, application hang, etc.) for Office client applications such as Word, Excel, PowerPoint, and Outlook. We use Ungraceful Application Exit metrics to measure the health of Office client products. It is a business-critical signal used to infer product stability.

The following fields are collected:

- **AffectedProcessAppBuild** - Build version identifier for the affected process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AffectedProcessAppBuildRevision** - Build revision identifier for the affected process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AffectedProcessAppMajorVer** - Minor version identifier for the affected process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AffectedProcessAppMinorVer** - Minor version identifier for the affected process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AffectedProcessAppName** - The name of the affected process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AffectedProcessAppVersion** - Version identifier for the affected process.

- **AffectedProcessExeBuildVersion** - The Build Version number of the affected process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AffectedProcessExeMajorVersion** - The Major Version number of the affected process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AffectedProcessExeMinorVersion** - The Minor Version number of the affected process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AffectedProcessExeRevisionVersion** - The Build Revision Version number of the affected process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AffectedProcessIsDebug** - Whether the affected process is a debug build. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AffectedProcessIsLabMachine** - Whether the affected process is in a Microsoft lab. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AffectedProcessOsEnvironment** - An operating system identifier for the affected process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AppName** - The name of the affected application. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AppUsedVirtualMemory** - Virtual memory utilized by office application
- **BucketId** - Watson bucket Identifier for crash
- **CabGuid** - GUID identifier for the Watson cab.
- **CallStack** - The Microsoft internal call stack causing the crash.
- **CrashedAssignedFlights** - The flights assigned to the crashed process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **CrashedConfigIds** - The configuration assigned to the crashed process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **CrashedEcsETag** - An experiment identifier for the crashed process.
- **CrashedImpressionId** - The impression identifier of the crashed process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **CrashedModuleName** - Failing module name
- **CrashedProcessSessionID** - A unique identifier of the crashed process.
- **CrashedProcessSessionInitTime** - The time when the affected process started.
- **CrashedProcessSessionUninitTime** - The time when the affected process ended.
- **CrashTag** - The unique identifier for the code of the crash.
- **CrashType** - Bucketing identifier for the type of crash.
- **DetectionTime** - The time when the unexpected exit was detected. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **ErrorString** - Error description. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **ExceptionAddress** - Address in the program where the failure occurred. *[This field has been removed*

from current builds of Office, but might still appear in older builds.]

- **ExceptionCode** - Bucketing identifier for the exception.
- **ExceptionInfo** - System information for the exception.
- **FaultAppName** - The name of the faulting app. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **HangTypeCode** - Represents class of hang if the process hung during execution.
- **InstallMethod** - Whether the current build of Office was upgraded from, rolled back to, or a fresh install.
- **InstallType** - An identifier for the method by which Office was installed. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **InstallTypeName** - An identifier for the method by which Office was installed. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **IsLabMachine** - Whether Office is being run in a Microsoft lab. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **IsMsftInternal** - Whether the Windows user running Office is a Microsoft employee. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **ModuleBaseAddress** - Base Address of the failing module. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **ModuleBuildVersion** - Failing module build version number. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **ModuleMajorVersion** - Failing module major version number. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **ModuleMinorVersion** - Failing module minor version number. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **ModuleName** - Failing module name. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **ModuleOffset** - Offset in bytes (in hexadecimal) from the base address where the failure occurred.
- **ModuleRevisionVersion** - Failing module build revision version number. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **ModuleSize** - Failing module size in bytes. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **ModuleVersion** - Version of the fault module responsible for a crash.
- **OfficeArchitectureText** - The architecture of the install: x64, x86, etc.
- **OfficeUILang** - The Language of the User Interface in the Office Build.
- **OSEnvironment** - Identifier for what environment Office is running on.
- **PreviousBuild** - Previously installed build version
- **ProcessorArchitecture** - Processor Architecture for the environment: x64, x86, etc.
- **SessionFlags** - Defines the conditions of the session such as: was file opened, or edited, was cloud document opened, was boot sequence completed, etc.

- **StackHash** - Provides a hashed id for the failure stack in Office.
- **SystemAvailableMemory** - Available memory in the operating system
- **UAETypeName** - Bucketing identifier for how the app exited ungracefully. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **UninitLibletId** - The unique identifier for the failing component of the crash.
- **VerifyElseCrashTag** - Unique identifier for where the app crashed. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **WatsonReportId** - Identifier of report sent to the Windows Watson service.
- **WerEventCreatedTime** - Time stamp for Windows Error Reporting event.

Office.System.SystemHealthUngracefulAppExitImmersive

Used to capture crash metrics.

The following fields are collected:

- **AffectedProcessAppBuild** - Build version identifier for the affected process.
- **AffectedProcessAppBuildRevision** - Build revision identifier for the affected process.
- **AffectedProcessAppMajorVer** - Major version identifier for the affected process.
- **AffectedProcessAppMinorVer** - Minor version identifier for the affected process.
- **AffectedProcessAppName** - The name of the affected process.
- **AffectedProcessExeBuildVersion** - The Build Version number of the affected process.
- **AffectedProcessExeMajorVersion** - The Major Version number of the affected process.
- **AffectedProcessExeMinorVersion** - The Minor Version number of the affected process.
- **AffectedProcessExeRevisionVersion** - The Build Revision Version number of the affected process.
- **AffectedProcessIsDebug** - Whether the affected process is a debug build.
- **AffectedProcessIsLabMachine** - Whether the affected process is in a Microsoft lab.
- **AffectedProcessOsEnvironment** - An operating system identifier for the affected process.
- **AppName** - The name of the affected application.
- **CrashedAssignedFlights** - The flights assigned to the crashed process.
- **CrashedConfigIds** - The configuration assigned to the crashed process.
- **CrashedImpressionId** - The impression identifier of the crashed process.
- **CrashedInteractionSessionID** - The interaction session identifier for the affected process.
- **CrashedInteractionSessionTime** - The time when the affected process could be interacted with.
- **CrashedProcessSessionID** - A unique identifier of the crashed process.
- **CrashedProcessSessionInitTime** - The time when the affected process started.
- **DetectionTime** - The time when the unexpected exit was detected.
- **IsLabMachine** - Whether Office is being run in a Microsoft lab.

- **IsMsftInternal** - Whether the Windows user running Office is a Microsoft employee.
- **OSEnvironment** - Identifier for what environment Office is running on.
- **PreviousLifecycleState** - The state of the affected process when it crashed.
- **UAETypeName** - Bucketing identifier for how the app exited ungracefully.

Office.System.SystemHealthUngracefulApplicationExitWin32

The event is triggers by an abnormal application termination (for example, task manager kill, application hang, etc.) for Office client applications such as, but not limited to, Word, Excel, PowerPoint, and Outlook. We use Ungraceful Application Exit metrics to measure the health of Office client products. It is a business-critical signal used by Office engineers to infer product stability.

The following fields are collected:

- **AddinExecution** - Flag that informs if an add-in was executing and didn't finish during an ungraceful application exit. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **AppUsedVirtualMemory** - Virtual memory utilized by office application
- **BootCompleted** - Was Office boot completed at the time of crash. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **BucketId** - Watson bucket Identifier for crash
- **CabGuid** - Globally Unique Identifier (GUID) identifier for the Watson cab.
- **CallStack** - The Microsoft internal call stack causing the crash.
- **CrashedAppBuild** - Build version identifier for the affected process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **CrashedAppMajor** - Major version identifier for the affected process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **CrashedAppMinor** - Minor version identifier for the affected process. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **CrashedAppVersion** - Application version identifier for crashed process.
- **CrashedEcsETag** - An experiment identifier for the crashed process.
- **CrashedModuleName** - Failing module name.
- **CrashedProcessSessionId** - A unique identifier of the crashed process.
- **CrashedProcessSessionInitTime** - The time when the affected process started.
- **CrashedProcessSessionUninitTime** - The time when the affected process ended.
- **CrashTag** - The unique identifier for the code of the crash.
- **CrashTime** - The time indicating the client terminated ungracefully. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **CrashType** - Bucketing identifier for the type of crash.
- **DetectionTime** - The time when the unexpected exit was detected. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **ExceptionAddress** - Address in the program where the failure occurred. *[This field has been removed*

from current builds of Office, but might still appear in older builds.]

- **ExceptionCode** - Bucketing identifier for the exception.
- **ExceptionInfo** - System information for the exception.
- **HandOff** - Did the user create and hand off the Office process to a new session. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **HangTypeCode** - Represents class of hang if the process hung during execution.
- **HasEdit** - Was the user editing a document in the crashed client. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **HasOpen** - Was a document open in the crashed client. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **HexCrashTag** - The unique identifier for the code of the crash. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **HexExceptionAddress** - Address in hexadecimal in the program where the failure occurred. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **HexExceptionCode** - Bucketing identifier in hexadecimal for the exception. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **HexModuleBaseAddress** - Base Address in hexadecimal of the failing module. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **HexModuleOffset** - Offset in bytes (in hexadecimal) from the base address where the failure occurred. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **HexModuleSize** - Failing module size in bytes in hexadecimal. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **HexVerifyElseCrashTag** - Unique identifier in hexadecimal for where the app crashed. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **InstallMethod** - Whether the current build of Office was upgraded from, rolled back to, or a fresh install.
- **IsLabMachine** - Whether Office is being run in a Microsoft lab. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **ModuleBaseAddress** - Base Address of the failing module. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **ModuleOffset** - Offset in bytes (in hexadecimal) from the base address where the failure occurred.
- **ModuleSize** - Failing module size in bytes. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **ModuleStamp** - Failing module stamp.
- **ModuleVersion** - Version of the fault module responsible for a crash.
- **OfficeArchitectureText** - The architecture of Office product as string (e.g. x86, arm).
- **OfficeUILang** - The language of the User Interface in the Office build.
- **PreviousBuild** - Previously installed build version
- **ProcessorArchitecture** - Processor Architecture for the environment x64, x86, etc.

- **SafeMode** - Was the session booted in safe mode. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **SessionFlags** - Defines the conditions of the session such as: was file opened, or edited, was cloud document opened, was boot sequence completed, etc.
- **StackHash** - Provides a hashed id for the failure stack in Office.
- **SystemAvailableMemory** - Available memory in the operating system
- **UAEOSEnvironment** - Operating System environment identifier. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **UninitLibletId** - The unique identifier for the failing component of the crash.
- **VerifyElseCrashTag** - Unique identifier for where the app crashed. *[This field has been removed from current builds of Office, but might still appear in older builds.]*
- **WatsonReportId** - Identifier of report sent to the Windows Watson service.
- **WerEventCreatedTime** - Time stamp for Windows Error Reporting event.

Office.System.UngracefulApplicationExit.DesktopAppExit

Used to capture crash metrics.

The following fields are collected:

- **AppBuildVersion** - Build version identifier for the affected process.
- **AppMajorVersion** - Major Version number of the affected process.
- **AppMinorVersion** - Minor version identifier for the affected process.
- **AppName** - The name of the affected application.
- **AppRevisionVersion** - Build revision identifier for the affected process.
- **CrashedAssignedFlights** - The flights assigned to the crashed process.
- **CrashedConfigIds** - The configuration assigned to the crashed process.
- **CrashedImpressionId** - The impression identifier of the crashed process.
- **CrashedInteractionSessionId** - The interaction session identifier of the crashed process.
- **CrashedProcessSessionId** - A unique identifier of the crashed process.
- **CrashType** - Bucketing identifier for the type of crash.
- **ErrorString** - Error description.
- **ExceptionAddress** - Address in the program where the failure occurred.
- **ExceptionCode** - Bucketing identifier for the exception.
- **FaultAppName** - The name of the faulting app.
- **InstallMethod** - Whether the current build of Office was upgraded from, rolled back to, or a fresh install.
- **InstallType** - An identifier for the method by which Office was installed.
- **IsDebug** - Whether this is a debug build of Office.
- **IsHandledCrash** - Whether the crash handler was invoked in the crashing session.

- **IsLabMachine** - Whether Office is being run in a Microsoft lab.
- **ModuleBaseAddress** - Base Address of the failing module.
- **ModuleName** - Failing module name.
- **ModuleOffset** - Offset in bytes from the base address where the failure occurred.
- **ModuleSize** - Failing module size in bytes.
- **OSEnvironment** - Identifier for what environment Office is running on.
- **PreviousBuild** - Previously installed build version
- **PreviousInteractionSessionTime** - Time which the previous interaction session started.
- **PreviousLifecycleState** - Previous session lifecycle state identifier.
- **PreviousSessionInitTime** - Time when the previous session started.
- **StackHash** - An identifier indicating where in code that the affected process crashed.
- **VerifyElseCrashTag** - Unique identifier for where the app crashed.

Office.System.UserChangedDiagnosticLevel

Information required to insure user privacy policy choices are being enforced.

The following fields are collected:

- **DiagnosticLevelChanged**: Indicates that the user changed their diagnostic level.
- **NewDiagnosticLevel**: The level after the user change.
- **OldDiagnosticLevel**: The level the user was using prior to their change.

Office.Telemetry.AriaEventSink.HandleMsaDeviceTokenResponse

Signal of an outage in Microsoft Account service.

The following fields are collected:

- **RetryCount** - Number of reties connecting to the MSA service.

Office.Telemetry.AriaEventSink.RequestMsaDeviceToken

Signal of an outage in Microsoft Account service.

The following fields are collected:

- **RetryCount** - Number of reties connecting to the Microsoft account service.

Office.Telemetry.ClientSamplingOverridden

Required to get reproduction rates right. Normally doesn't apply to Production audience group.

The following fields are collected:

- **OverriddenMeasureEnabled** - Is the client set to send more than unsampled events
- **OverriddenNumberlinePosition** - The new number line position for sampling
- **OverriddenReportedSampleRate** - The new reported sample rate
- **OverriddenSampleRate** - The new sample rate
- **PreviousNumberlinePosition** - The sampling position on the number line.

- **PreviousSampleRate** - The sample rate prior to being overridden.
- **WasMeasureEnabled** - Was the client set to send more than unsampled events

Office.Telemetry.Compliance.EventNotInBasicAllowList

Reports invalid telemetry implementations or deployments

The following fields are collected:

- **EventName** - The name of the event that is not in the list

Office.Telemetry.Compliance.MissingDataCategory

Reports invalid telemetry implementations or deployments

The following fields are collected:

- **EventName** - Event name that is missing a category
- **IsFromRule** - Whether the event came from a telemetry rule

Office.Telemetry.Compliance.MissingDataCategoryInRule

Reports invalid telemetry implementations or deployments

The following fields are collected:

- **RuleId** - The rule ID that is missing a data category
- **RuleVersion** - The rule version that is missing a data category

Office.Telemetry.DiagnosticDataViewerStateChanged

Validates that consumers can view the data as it leaves their machine using the Diagnostic Data Viewer.

The following fields are collected:

- **Dialogcanceled** - Was the diagnostic data viewer dialog canceled
- **NewState** - New diagnostic data viewer state
- **WasDialogUsed** - Was the diagnostic data viewer dialog used

Office.Telemetry.DynamicConfig.FetchConfigs

Data needed to measure health of Telemetry Config Service.

The following fields are collected:

- **ParsedConfigCount** - Number of parsed dynamic configs
- **ParsedConfigs** - Number of parsed dynamic configs
- **RejectedConfigCount** - Number of rejected configs
- **RejectedConfigs** - Number of rejected configs
- **RejectedConfigsList** - Comma-separated list of rejected configs.

Office.Telemetry.DynamicConfig.ParseJsonConfig

Data needed to measure the health of Telemetry Config Service

The following fields are collected:

- **ErrorMessage** - Parsing error message
- **NodeName** - Node which failed to parse

Office.Telemetry.DynamicConfig.PopulatedRequestIgnored

This event is generated when we fail to set up the telemetry configuration pipeline.

This event collects no fields.

Office.Telemetry.DynamicConfig.PopulateTreeCalledAgain

Data needed to measure health of Telemetry Config Service.

This event collects no fields.

Office.Telemetry.EventQuarantined

Used to verify other NSD events are working properly.

The following fields are collected:

- **EventName** - Quarantined event name
- **Reason** - Reason for quarantine

Office.Telemetry.FlushEventBuffer

Reports event buffer size and can indicate telemetry failures related to large buffer use.

The following fields are collected:

- **EventCount** - Count of events in the buffer
- **FirstPassCount** - First pass count of events
- **SecondPassCount** - Second pass count of events

Office.Telemetry.GetFilteredPayloadsFromDisk

Verifies certain parts of legacy telemetry pipeline are working on platforms that still use it.

This event collects no fields.

Office.Telemetry.InvalidDataContractName

Reports invalid telemetry implementations or deployments

The following fields are collected:

- **DataContractName** - Name of the telemetry data contract
- **EventName** - Name of the event with the invalid data contract
- **IsRuleEvent** - True/false was this event implemented by a telemetry rule

Office.Telemetry.InvalidDataFieldName

Reports invalid telemetry implementations or deployments

The following fields are collected:

- **DataFieldName** - Name of the telemetry data field
- **EventName** - Name of the event with the invalid field
- **IsRuleEvent** - True/false was this event implemented by a telemetry rule.

Office.Telemetry.InvalidEventContractName

Reports invalid telemetry implementations or deployments

The following fields are collected:

- **EventContractName** - The invalid telemetry contract name
- **EventName** - Name of the event with the invalid contract name
- **IsRuleEvent** - True/false was this event implemented by a telemetry rule

Office.Telemetry.LoadXmlRules

Reports whether parsing telemetry rules succeeded

The following fields are collected:

- **DetachedDuration** - Detached duration in microseconds

Office.Telemetry.MissingFieldDetails

Reports missing field information to diagnose typos in telemetry configuration.

The following fields are collected:

- **ErrorRuleId** - The telemetry rule ID that requested the missing field
- **ErrorRuleVersion** - The telemetry rule version that requested the missing field
- **EtwEventGuid** - The ETW GUID of the requested field
- **EtwEventId** - The ETW event ID of the requested field
- **MissingFieldName** - The requested field name
- **UlsTagId** - The code tag of the missing field

Office.Telemetry.ProcessIdleQueueJob

Reports that telemetry idle processing started as expected.

The following fields are collected:

- **DetachedDuration** - Detached duration in microseconds
- **FailureDiagnostic** - The failed operation

Office.Telemetry.RedstoneInboxSampling

Sampling state of the client required to accurately interpret other metrics.

The following fields are collected:

- **MeasuresEnabled** - Are measures enabled in this session?
- **SamplingClientIdValue** - Sampling value for this client
- **SamplingKey** - Sampling key for this client
- **SamplingMethod** - Sampling method for this client

Office.Telemetry.RedstoneInboxSamplingCritical

Sampling state of the client can be required to accurately interpret other metrics.

The following fields are collected:

- **MeasuresEnabled** - Are measures enabled in this session?
- **SamplingClientIdValue** - Sampling value for this client
- **SamplingKey** - Sampling key for this client
- **SamplingMethod** - Sampling method for this client

Office.Telemetry.RuleErrorsAggregated

Telemetry health error reporting. Required to validate other data (including NSD).

The following fields are collected:

- **ErrorCount** - Count of this error within the aggregation time window
- **ErrorInfo** - Error diagnostic info number
- **ErrorRuleId** - Telemetry rule ID that caused the error
- **ErrorRuleVersion** - Telemetry rule version that caused the error
- **WarningInfo** - Warning diagnostic info number
- **QueueFlushCount** - Number of queue flushes
- **QueueFlushDueToSizeLimit** - Size at which telemetry flushes the queue
- **QueueFlushesDueToSize** - Count of queue flushes caused by buffer size
- **QueueHardLimit** - Telemetry shutdown limit
- **QueueLimitHitTime** - When the shutdown limit was reached
- **ResultTime** - Time of this event

Office.Telemetry.RulesEngineDiskThrottled

Throttling DQ metrics. Required for confidence in all other data.

The following fields are collected:

- **DiskWriteLimit** - Disk size limit for telemetry data
- **DiskWriteTotal** - Disk write total for telemetry data
- **SessionDiskWriteTotal** - Session disk write total for telemetry data
- **ThrottlingTimestamp** - Time the session was throttled

Office.Telemetry.RulesEngineMediumCostThrottled

Throttling DQ metrics. Required for confidence in all other data.

This event collects no fields.

Office.Telemetry.RulesEngineSpikeThrottled

Throttling DQ metrics. Required for confidence in all other data.

The following fields are collected:

- **CurrentLimit** - Current spike limit
- **Duration** - Spike duration
- **Factor** - Spike factor
- **HighestImpactingRuleBytes** - The most bytes recorded by a telemetry rule
- **HighestImpactingRuleId** - The rule ID that recorded the most bytes
- **HighestImpactingRuleVersion** - The rule version that recorded the most bytes
- **MaxLimit** - The maximum limit

- **ThrottlingTimestamp** - When telemetry was throttled

Office.Telemetry.RulesEngineThrottled

Throttling DQ metrics. Required for confidence in all other data.

The following fields are collected:

- **ThrottlingTimestamp** - When telemetry was throttled

Office.Telemetry.RulesEngineUlsQueueSizeBackgroundProcessingLevelReached

Reports that there are too many events in the queue to process during app idle time.

The following fields are collected:

- **BackgroundProcessingLevelInBytes** - The queue size to start processing in the background.
- **CurrentQueueSize** - The number of events in the nULS queue.
- **CurrentQueueSizeInBytes** - The size of the nULS queue in bytes.
- **ReachedTimestamp** - The time when background processing began.

Office.Telemetry.RulesResultUploadLatencyRule

The Average, Min and Max upload Latency of rule results payload upload every hour

The following fields are collected:

- **AverageLatency** - The average upload latency.
- **CollectionTime** - The time when data on rule upload was collected.
- **LatencyGE201LE400** - The number of uploads with a latency greater than or equal to 201ms and less than or equal to 400ms
- **LatencyGE3001** - The number of uploads with a latency greater than or equal to 3001ms.
- **LatencyGE401LE600** - The number of uploads with a latency greater than or equal to 401ms and less than or equal to 600ms.
- **LatencyGE601LE800** - The number of uploads with a latency greater than or equal to 601ms and less than or equal to 800ms.
- **LatencyLE200** - The number of uploads with a latency less than 200 milliseconds.
- **MaxLatency** - The highest latency observed.
- **MinLatency** - The lowest latency observed.

Office.Telemetry.SamplingPolicy

Sampling state of the client required to accurately interpret other metrics.

The following fields are collected:

- **MeasuresEnabled** - Are measures enabled in this session?
- **SamplingClientIdValue** - Sampling value for this client
- **SamplingKey** - Sampling key for this client
- **SamplingMethod** - Sampling method for this client

Office.Telemetry.SamplingPolicyEventTrigger

Sampling state of the client required to accurately interpret other metrics.

The following fields are collected:

- **MeasuresEnabled** - Are measures enabled in this session?
- **SamplingKey** - Sampling key for this client
- **SamplingMethod** - Sampling method for this client

Office.Telemetry.SessionTelemetryRulesChanged

Reports that the set of telemetry rules has changed

The following fields are collected:

- **ChangedRuleId** - The telemetry rule ID that changed in the current update
- **ChangedRuleVersion** - The telemetry rule version that changed in the current update
- **OperationType** - Add or remove operation tag

Office.Telemetry.SessionTelemetryRulesCount

Reports the count of loaded telemetry rules

The following fields are collected:

- **CountOfLoadedRules** - How many telemetry rules are loaded
- **HadRuleFileAtBoot** - Whether there was a telemetry rules file at app boot

Office.Telemetry.SessionTelemetryRulesInitialState

Reports the telemetry rules that were loaded at session start

The following fields are collected:

- **HadRuleFileAtBoot** - Whether there was a telemetry rules file at app boot
- **LoadedRulesCount** - How many telemetry rules are loaded
- **LoadedRulesList** - List of loaded telemetry rules

Office.Telemetry.SystemHealthMetadataNetworkCost

Network cost indicates our ability to get data or not.

The following fields are collected:

- **NetworkCost** - New network metered or unmetered cost
- **OldNetworkCost** - Previous network metered or unmetered cost
- **Tag** - Source code tag that detected the change

Office.Telemetry.SystemHealthMetadataNetworkCostChange

Network cost indicates our ability to get data or not.

The following fields are collected:

- **NewNetworkCost** - New network metered or unmetered cost
- **OldNetworkCost** - Previous network metered or unmetered cost
- **Tag** - Source code tag that detected the change

Office.Telemetry.TelemetryActivityAggregationWindowStatistics

Reports the number of aggregated activity groups and instances in each activity being uploaded.

The following fields are collected:

- **GroupCount** - The number of aggregated activities sending data.
- **InstancesToSend** - The number of instances of aggregated Activities sending data.

Office.Telemetry.TelemetryUlsQueueUsage

Telemetry health error reporting. Required to validate other data (including NSD).

The following fields are collected:

- **AverageEventCount** - Average event count in the queue
- **AverageQueueCB** - Average memory size of the queue
- **PeakEventCount** - Peak event count of the queue
- **PeakQueueCB** - Peak memory size of the queue
- **QueueDisableRuleLimit** - Limit at which telemetry rules get disabled

Office.Telemetry.UlsQueueTopThrottlingTags

Reports the top tags that contributed to ULS queue being shut down.

The following fields are collected:

- **Tag0** - Tag which consumed the most queue
- **Tag0Percent** - Percentage of queue used by tag0
- **Tag1** - Tag which consumed the 2nd highest amount of queue
- **Tag10** - Tag which consumed the 11th highest amount of queue
- **Tag10Percent** - Percentage of queue used by tag10
- **Tag11** - Tag which consumed the 12th highest amount of queue
- **Tag11Percent** - Percentage of queue used by tag11
- **Tag12** - Tag which consumed the 13th highest amount of queue
- **Tag12Percent** - Percentage of queue used by tag12
- **Tag13** - Tag which consumed the 14th highest amount of queue
- **Tag13Percent** - Percentage of queue used by tag13
- **Tag14** - Tag which consumed the 15th highest amount of queue
- **Tag14Percent** - Percentage of queue used by tag14
- **Tag1Percent** - Percentage of queue used by tag1
- **Tag2** - Tag which consumed the third highest amount of queue
- **Tag2Percent** - Percentage of queue used by tag2
- **Tag3** - Tag which consumed the fourth highest amount of queue
- **Tag3Percent** - Percentage of queue used by tag3
- **Tag4** - Tag which consumed the fifth highest amount of queue
- **Tag4Percent** - Percentage of queue used by tag4

- **Tag5** - Tag which consumed the sixth highest amount of queue
- **Tag5Percent** - Percentage of queue used by tag5
- **Tag6** - Tag which consumed the seventh highest amount of queue
- **Tag6Percent** - Percentage of queue used by tag6
- **Tag7** - Tag which consumed the eighth highest amount of queue
- **Tag7Percent** - Percentage of queue used by tag7
- **Tag8** - Tag which consumed the ninth highest amount of queue
- **Tag8Percent** - Percentage of queue used by tag8
- **Tag9** - Tag which consumed the tenth highest amount of queue
- **Tag9Percent** - Percentage of queue used by tag9

Office.Telemetry.VolumeTrackingData

Event volume tracking metrics for telemetry events

The following fields are collected:

- **EventThreshold** - The maximum number of instances of a single event that can be sent in a window of time.
- **HighestEventCount** - The highest number of instances of a single event sent this window.
- **HighestEventName** - The name of the event with the highest number of instances in this window.
- **TimeWindowInSeconds** - The duration of the window in seconds.
- **TotalEvents** - The total number of events sent during the window.
- **UniqueEvents** - The number of unique events sent during a window.

Office.Telemetry.WritePayloadsToDisk

Verifies certain parts of legacy pipeline are working on platforms that still use it.

The following fields are collected:

- **DetachedDuration** - Detached duration in microseconds

In-product recommendations in Office

8/25/2021 • 2 minutes to read • Edit Online

To help you take advantage of the features and functionality of Office, we might provide recommendations to you within an Office product that you're using. For organizational users, these recommendations may include messages related to productivity products or services that your organization has purchased or licensed, whether or not you're currently using those products or services. For consumer users, these recommendations may also include messages related to productivity products or services that are available to you and that are free to use.

Here are some examples of the types of in-product recommendations that you might see:

- A recommendation to use PowerPoint Designer to provide you with design ideas as you're creating slides for a presentation.
- A recommendation to save your document on OneDrive so that you can easily access the document from other devices.
- A recommendation to use Excel for iOS or Android so that you can access your files from your mobile device.

Our goal is to provide recommendations that are relevant, timely, and shown in the appropriate context. To provide these recommendations, we might rely on information about the Office products you have purchased, the Office apps that you use, the features and capabilities within those Office apps that you use, or those apps that you're licensed to use but haven't tried yet. These recommendations are not based on the actual content that you create while using Office, such as budget projections in an Excel spreadsheet or the text you write in an Outlook email.

For more information, see Privacy at Microsoft.