

An examination of Win10 ActivitiesCache.db database

Windows Timeline, is a new feature of Windows 10 introduced with version 1803.
It is part of the Connected Devices Platform

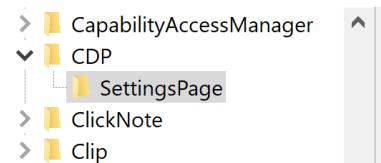
- The [Connected Devices Platform Service](#), is a Windows service that provides a way for devices such as PC's and smartphones to discover and send messages between each other.
- Connected Devices [Platform Service \(CDPSvc\) Defaults](#) in Windows 10.

and the Microsoft Graph's Cross-device experience ([Project Rome](#)) .

The CDP settings for the Current User are stored in the registry at:
NTUSER.DAT -> 'Software\Microsoft\Windows\CurrentVersion\CDP'

Name	Type	Data
ab(Default)	REG_SZ	(value not set)
8110 CdpSessionUserAuthzPolicy	REG_DWORD	0x00000001 (1)
ab CdpUserSettingsVersion	REG_SZ	RS4
8110 EnableRemoteLaunchToast	REG_DWORD	0x00000001 (1)
8110 NearShareChannelUserAuthzPolicy	REG_DWORD	0x00000000 (0)
8110 RomeSdkChannelUserAuthzPolicy	REG_DWORD	0x00000001 (1)

and

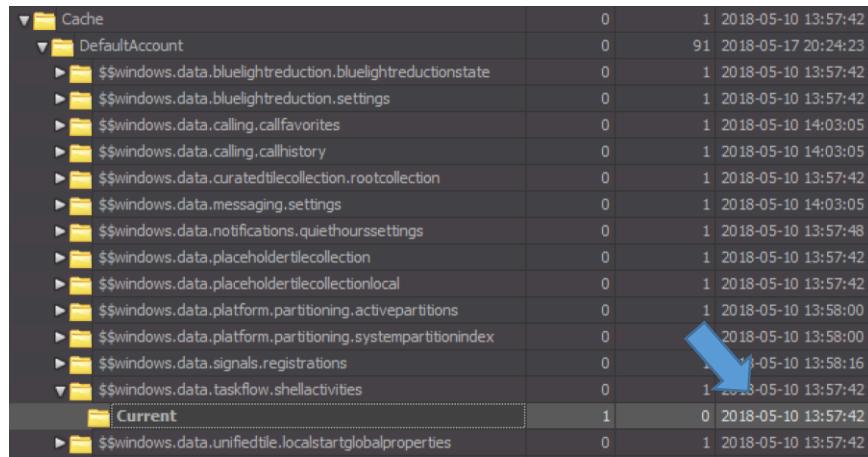


Name	Type	Data
ab(Default)	REG_SZ	(value not set)
8110 BluetoothLastDisabledNearShare	REG_DWORD	0x00000000 (0)
8110 NearShareChannelUserAuthzPolicy	REG_DWORD	0x00000001 (1)

Before Windows 10 version 1803

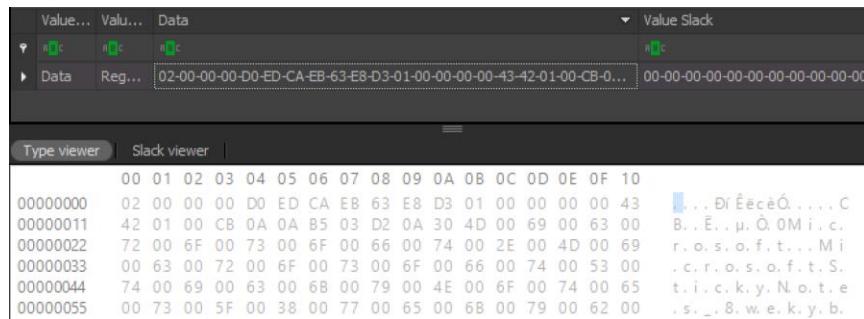
The service and the 'ActivitiesCache.db' database existed before the 1803 upgrade (May 2018), but with limited functionality. Another possibly related* activity store location is at:
'Software\Microsoft\Windows\CurrentVersion\CloudStore\Store\Cache\DefaultAccount\\$\\$windows.data.taskflow.shellactivities\Current'

*(considering that the current ActivitiesCache.db uses taskflow to retrieve device information as seen further below in this document)



		0	1	2018-05-10 13:57:42
▼	Cache	0	91	2018-05-17 20:24:23
▼	DefaultAccount	0	1	2018-05-10 13:57:42
►	\$\$windows.data.bluelightreduction.bluelightreductionstate	0	1	2018-05-10 13:57:42
►	\$\$windows.data.bluelightreduction.settings	0	1	2018-05-10 13:57:42
►	\$\$windows.data.calling.callfavorites	0	1	2018-05-10 14:03:05
►	\$\$windows.data.calling.callhistory	0	1	2018-05-10 14:03:05
►	\$\$windows.data.curatedtilecollection.rootcollection	0	1	2018-05-10 13:57:42
►	\$\$windows.data.messaging.settings	0	1	2018-05-10 14:03:05
►	\$\$windows.data.notifications.quiethourssettings	0	1	2018-05-10 13:57:48
►	\$\$windows.data.placeholdertilecollection	0	1	2018-05-10 13:57:42
►	\$\$windows.data.placeholdertilecollectionlocal	0	1	2018-05-10 13:57:42
►	\$\$windows.data.platform.partitioning.activepartitions	0	1	2018-05-10 13:58:00
►	\$\$windows.data.platform.partitioning.systempartitionindex	0	1	2018-05-10 13:58:00
►	\$\$windows.data.signals.registrations	0	1	2018-05-10 13:58:16
▼	\$\$windows.data.taskflow.shellactivities	0	1	2018-05-10 13:57:42
►	Current	1	0	2018-05-10 13:57:42
►	\$\$windows.data.unifiedtile.localstartglobalproperties	0	1	2018-05-10 13:57:42

where there is a value named 'Data':



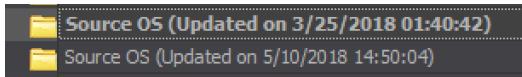
Value...	Value...	Data	Value	Slack
02 00 00 00 D0 ED CA E8 63 E8 D3 01 00 00 00 43 42 01 00 CB 0...	Reg...	02-00-00-00-D0-ED-CA-E8-63-E8-D3-01-00-00-00-43-42-01-00-CB-0...	00-00	

Type viewer | Slack viewer |

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
00000000	02	00	00	00	D0	ED	CA	E8	63	E8	D3	01	00	00	00	43
00000011	42	01	00	CB	0A	0A	B5	03	D2	0A	30	4D	00	69	00	63
00000022	72	00	6F	00	73	00	6F	00	66	00	74	00	2E	00	4D	00
00000033	00	63	00	72	00	6F	00	73	00	6F	00	66	00	74	00	53
00000044	74	00	69	00	63	00	6B	00	79	00	4E	00	6F	00	74	00
00000055	00	73	00	5F	00	38	00	77	00	65	00	6B	00	79	00	62

Its value is in hex and it seems to hold interesting information, including the Filetime of last update (which corresponds to the Last Write Timestamp of the registry key).

If Windows is updated to version 1803, this 'log' stops being updated. This can be checked by looking in the SYSTEM hive at the Setup key like:



In that case, interestingly,

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI ASCII							
000000000	02	00	00	00	D0	ED	CA	EB	63	E8	D3	01	00	00	00	00	Điề́cè́o							
000000016	43	42	01	00	CB	0A	0A	B5	03	FILETIME: 10-05-18 16:36:37 +3:0				CB	È	µ	Ò	OM	i					
000000032	63	00	72	00	6F	00	73	00	6F	00	66	00	74	00	2E	00	c	r	o	s	o	f	.	
000000048	4D	00	69	00	63	00	72	00	6F	00	73	00	6F	00	66	00	M	i	c	r	o	s	o	f
000000064	74	00	53	00	74	00	69	00	63	00	6B	00	79	00	4E	00	t	s	t	i	c	k	y	N
000000080	6F	00	74	00	65	00	73	00	5F	00	38	00	77	00	65	00	o	t	e	s	_	8	w	e

this Filetime is very close to the date of the ntuser.dat.LOG files (*which coincides with the date the 1803 update occurred*), and that can also be seen from the last entry above:

NTUSER.DAT	12 Jun 18 1:06 pm
ntuser.dat.LOG1	10 May 18 4:50 pm
ntuser.dat.LOG2	10 May 18 4:50 pm

Further [examination](#) shows a consistent pattern:

- 0xD2 14 = Start of Entry
- Next byte = length of block (x2)
- Start of path & executable
- 0xC6 1F = End of block
- Next 4 bytes = unknown
- 0xD2 23 = Executable Block
- Next byte = length of block (x2)
- Executable
- 0xD2 28 = Payload block
- Next byte = length of block (x2)
- Payload (eg email, URL etc.)
- 0xC6 32 = end of block
- Next 9 bytes = (A) is the same as (B) of the next entry (upwards)
- 0xC6 3C = Pointer to next entry
- Next 9 bytes = (B) is the same as (A) of the next entry (upwards) *
- 0xCA500000 = End of Entry

*Top most entry is the newest one, so for the last entry these 9 bytes are all 0xFF

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000000000	02	00	00	00	69	BC	79	90	8F	E9	D3	01	00	00	00	00
000000016	43	42	01	00	CB	0A	0A	91	03	D2	14	25	FILETIME: 12-05-18			
000000032	5C	00	24	00	57	00	49	00	4E	00	44	00	4F	00	57	00
000000048	53	00	2E	00	7E	00	42	00	54	00	5C	00	53	00	6F	00
000000064	75	00	72	00	63	00	65	00	73	00	5C	00	53	00	65	00
000000080	74	00	75	00	70	00	48	00	6F	00	73	00	74	00	2E	00
000000096	65	00	78	00	65	00	C6	1F	80	8D	DC	01	D2	23	0D	53
000000112	00	65	00	74	00	75	00	70	00	48	00	6F	00	73	00	74
000000128	00	2E	00	65	00	78	00	65	00	D2	28	10	57	00	69	00
000000144	6E	00	64	00	6F	00	77	00	73	00	20	00	31	00	30	00
000000160	20	00	53	00	65	00	74	00	75	00	70	00	C6	32	80	9E
000000176	9D	A9	F3	B1	FA	E9	01	C6	3C	FF	FF	FF	FF	FF	FF	FF
000000192	FF	FF	01	CA	50	00	00	D2	14	39	46	00	3A	00	5C	00
000000208	5F	00	46	00	6F	00	72	00	65	00	6E	00	73	00	69	00
000000224	63	00	20	00	54	00	6F	00	6F	00	6C	00	73	00	5C	00
000000240	64	00	6D	00	64	00	65	00	2D	00	70	00	72	00	6F	00
000000256	66	00	2D	00	32	00	21	00	34	00	2E	00	36	00	2E	00
000000272	34	00	34	00	38	00	2D	00	77	00	69	00	6E	00	36	00
000000288	34	00	2D	00	67	00	75	01	69	00	5C	00	64	00	6D	00
000000304	64	00	65	00	2E	00	65	00	78	00	65	00	C6	1F	8E	9D
000000320	9C	01	D2	23	08	64	00	6D	00	64	00	65	00	2E	00	65
000000336	00	78	00	65	00	D2	28	17	49	00	4D	00	44	00	45	00
000000352	20	00	33	00	2E	00	34	00	2E	00	32	00	20	00	46	00
000000368	72	00	65	00	65	00	20	00	45	00	64	00	69	00	74	00
000000384	69	00	6F	00	6E	00	C6	32	E7	81	B1	91	F3	B1	FA	E9
000000400	01	C6	3C	80	9E	9D	A9	F3	B1	FA	E9	01	CA	50	00	00
000000416	D2	14	2C	43	00	3A	00	5C	00	50	00	72	00	6F	00	67
000000432	00	72	00	61	00	6D	00	20	00	46	00	69	00	6C	00	65

With a bit of tweaking in Notepad++, it shows web page titles, email (used in accounts of Outlook), File Explorer paths followed and name of the remote devices accessed with Teamviewer among other.

```

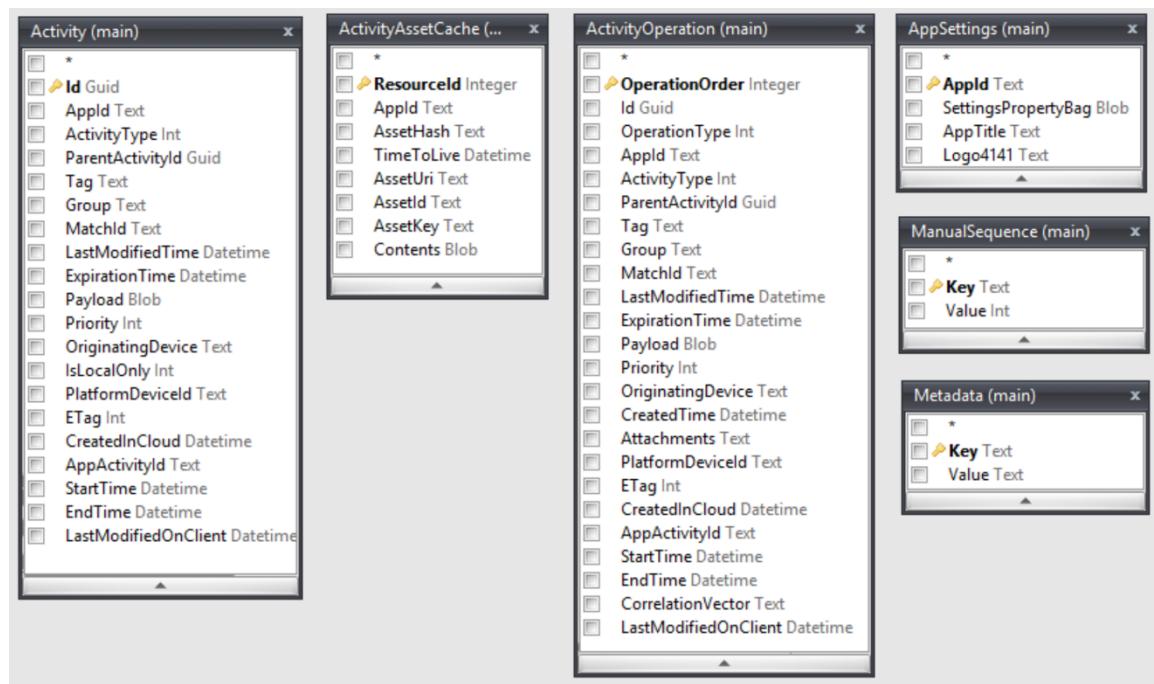
508
509 C:\Program Files (x86)\TeamViewer\TeamViewer.exe[08:50]#S TeamViewer.exe()
510 TeamViewerZ2Y'-t(S0H<9X'-t(S0HPC0X0)
511
512 C:\Program Files (x86)\TeamViewer\TeamViewer.exe[08:50]#S TeamViewer.exe() (Wait for partnerZ2Ap-t(S0Z<Y'-t(S0HPC0X0)
513
514 C:\Program Files (x86)\TeamViewer\TeamViewer.exe[08:50]#S TeamViewer.exe()
515 TeamViewerZ2B&Bf-t(S0Z<Ap-t(S0HPC0X0
516
517 C:\Program Files (x86)\TeamViewer\TeamViewer.exe[08:51:00]#S TeamViewer.exe() (Sponsored sessionZ2o"Y@0t(S0Z<B&Bf-t(S0HPC0X0
518
519 C:\Program Files (x86)\TeamViewer\TeamViewer.exe[08:50]#S TeamViewer.exe()
520 TeamViewerZ2QY00@t(S0Z<d"Y@0t(S0HPC0X0
521
522 C:\Program Files (x86)\TeamViewer\TeamViewer.exe[08:50:00]#S TeamViewer.exe() (Wait for partnerZ2o$E0@t(S0Z<QY00@t(S0HPC0X0
523
524 C:\Program Files (x86)\TeamViewer\TeamViewer.exe[08:51:00]#S TeamViewer.exe() (=ZEUS_X6 - TeamViewer - Free license (non-commercial use
only)Z2(Z@t(S0Z<E20@t(S0HPC0X0
525

```

Back to the 'ActivitiesCache.db' database.

The location of both old and new dBs is at the
"%userprofile%\appdata\local\ConnectedDevicesPlatform" folder.

The old dB table structure was similar to the new dB, but it included 6 tables + the master table (the 'Activity_PackageId' table was missing, and there were different fields):



The information held was also different:

E.g. the ‘AppSettings’ table in the old DB looked like this:

AppId	SettingsPropertyBag	AppTitle	Logo4141
1 windows.immersivecontrolpanel_cv5n1h2txyewy!microsoft.windows.immersivecontrolpanel	BLOB	NULL	NULL
2 Microsoft.SkypeApp_kzf8qxf38zg5c!ppleae38af2e007f4358a809ac99a64a67c1	BLOB	NULL	NULL
3 Microsoft.SkypeApp_kzf8qxf38zg5c!App	BLOB	NULL	NULL
4 Microsoft.MicrosoftEdge_8wekyb3d8bbwe!MicrosoftEdge	BLOB	NULL	NULL
5 AD2F1837.HPPrinterControl_l10z8vjag6ke6!AD2F1837.HPPrinterControl	BLOB	NULL	NULL
6 Microsoft.BingNews_8wekyb3d8bbwe!AppexNews	BLOB	NULL	NULL
7 Microsoft.WindowsCamera_8wekyb3d8bbwe!App	BLOB	NULL	NULL
8 Microsoft.Windows.Photos_8wekyb3d8bbwe!App	BLOB	NULL	NULL
9 VirtualPulse.PlayerforMedia_nh7p8cqfc4t04!App	BLOB	NULL	NULL

And included UWP ([Universal Windows Platform](#)) Apps only.

The new dB does not have any entries in this table.

The ‘Metadata’ table was also different:

Key	Value
1 CurrentEtag	5b15de40-92c2-11e7-9117-01020305070d
2 CurrentSettings	{"ActivityTypes": [4,3,2,1,0], "Environment": "prod"}
3 PendingStrongAuth	{"pendingStrongAuth": true}

Looking at the entries in an L.username folder and a MS account id folder, the username (local account) dB has these entries:

Key	Value
1 CurrentSettings	{"ActivityTypes": [4,3,2,1,0], "Environment": "prod"}

And the Microsoft Account dB these:

Key	Value
1 CurrentSettings	{"ActivityTypes": [4,3,2,1,0, 5,6], "Environment": "prod"}
2 CurrentEtag	12c3eb60-189a-11e8-9117-01020305070d

From this, we can deduce that ‘CurrentSettings’ is where the Activity Types of the entries populating the database are defined.

The Smartlookup view (query, included in the dB):

```
select
    [O].[Id],
    [O].[AppId],
    [O].[AppActivityId],
    [O].[ActivityType],
    [O].[ParentActivityId],
    [O].[Tag],
    [O].[Group],
    [O].[MatchId],
    [O].[LastModifiedTime],
    [O].[ExpirationTime],
    [O].[Payload],
    [O].[Priority],
    [O].[OriginatingDevice],
    [A].[IsLocalOnly],
    [A].[PlatformDeviceId],
    [A].[CreatedInCloud],
    [O].[StartTime],
    [O].[EndTime],
    [O].[LastModifiedOnClient],
    [O].[ETag]
from [ActivityOperation] as [O]
left outer join [Activity] as [A] on [O].[Id] = [A].[Id]
where [O].[OperationType] <> 3
union
select
    [Id],
    [AppId],
    [AppActivityId],
    [ActivityType],
    [ParentActivityId],
    [Tag],
    [Group],
    [MatchId],
    [LastModifiedTime],
    [ExpirationTime],
    [Payload],
    [Priority],
    [OriginatingDevice],
    [IsLocalOnly],
    [PlatformDeviceId],
    [CreatedInCloud],
    [StartTime],
    [EndTime],
    [LastModifiedOnClient],
    [ETag]
from [Activity]
where [Id] not in (select [Id]
                    from [ActivityOperation])
```

Typical ActivityOperation table entries were of ActivityType 2 :

Table:  ActivityOperation

Id	operationType	AppId	AppActivityId	ActivityType	ParentActivityId
1	1	{"alternateId": "Settings" ...}		2	BLOB
2	3	{"alternateId": "Settings" ...}		2	BLOB

and were in fact Notifications, similar to the ones we can see in
`%username%\AppData\Local\Microsoft\Windows\Notifications\wpndatabase.db`

The AppID was a json blob like this:

Tag	Value
object	
alternateId	Settings
instanceId	0
packageId	windows.immersivecontrolpanel_cw5n1h2tbyewy!microsoft.windows.immersivecontrolpanel
titleId	0
windows_universal	windows.immersivecontrolpanel_cw5n1h2tbyewy!microsoft.windows.immersivecontrolpanel

And the Payload field

Tag	Group	MatchId	lastModifiedTime	expirationTime	Payload
EditionUpgradeTag	EditionUpgradeGroup	NULL	1521896726	1522501525	<toast NotificationId="107" di...
EditionUpgradeTag	EditionUpgradeGroup	NULL	1521896726	1522501525	<toast NotificationId="107" di...

was an XML blob:

```

<toast NotificationId="107" displayTimestamp="131663703259843007" launch="page=SettingsPageActivate">
  <visual>
    <binding template="ToastText02">
      <text id="1">
        Success!
      </text>
      <text id="2">
        You're all done and your PC is ready to go.
      </text>
    </binding>
  </visual>
</toast>

```

From this, another deduction can be made, that ActivityType 2 is for Toast Notification entries.

The **new (1803)** dB has different fields:

Table: Metadata

Key	Value
1 CurrentSettings	{"ActivityTypes": [4,3,2,1,0,5,13,6], "Environment": "prod"}
2 DatabaseInstanceId	38608
3 DatabaseInstanceIdUpdateTime	2018-05-10T14:00:55.845Z

After examination of various new 'ActivitiesCache.db' files, any entries seen in the new database have an '**ActivityType**' (as *seen above*) of 5 or 6, which appear to represent '*Open App/File/Url*' and '*App In Focus*' respectively – type 13 entries were not observed in any dB yet.

The '**DatabaseInstanceIdUpdateTime**' date seen above, is the date & time (in UTC) that the current dB file was created.

A Microsoft Account related DB includes one more field, the '**CurrentEtag**' but instead of using an ETAG in the form found elsewhere in the dB, it lists it in the form of a GUID – that is the Cloud ETAG:

Key	Value
1 CurrentSettings	{"ActivityTypes": [4,3,2,1,0,5,13,6], "Environ..."}
2 DatabaseInstanceId	23919
3 DatabaseInstanceIdUpdateTime	2018-05-10T13:58:59.758Z
4 CurrentEtag	8df03b30-642c-11e8-9117-01020305070d

The ‘SmartLookUp’ view in the old DB was also different:

```
1 select      [O].[Id],  
2       [O].[AppId],  
3       [O].[AppActivityId],  
4       [O].[ActivityType],  
5       [O].[ParentActivityId],  
6       [O].[Tag],  
7       [O].[Group],  
8       [O].[MatchId],  
9       [O].[LastModifiedTime],  
10      [O].[ExpirationTime],  
11      [O].[Payload],  
12      [O].[Priority],  
13      [O].[OriginatingDevice],  
14      [A].[IsLocalOnly],  
15      [A].[PlatformDeviceId],  
16      [A].[CreatedInCloud],  
17      [O].[StartTime],  
18      [O].[EndTime],  
19      [O].[LastModifiedOnClient],  
20      [O].[ETag]  
21 from [ActivityOperation] as [O]  
22 left outer join [Activity] as [A] on [O].[Id] = [A].[Id]  
23 where [O].[OperationType] <> 3  
24 union  
25 select      [Id],  
26       [AppId],  
27       [AppActivityId],  
28       [ActivityType],  
29       [ParentActivityId],  
30       [Tag],  
31       [Group],  
32       [MatchId],  
33       [LastModifiedTime],  
34       [ExpirationTime],  
35       [Payload],  
36       [Priority],  
37       [OriginatingDevice],  
38       [IsLocalOnly],  
39       [PlatformDeviceId],  
40       [CreatedInCloud],  
41       [StartTime],  
42       [EndTime],  
43       [LastModifiedOnClient],  
44       [ETag]  
45 from [Activity]  
46 where [Id] not in (select [Id]  
47   from [ActivityOperation])
```

The new Timeline & ActivitiesCache.db

The **NEW** (updated) database stores information for each user in the %LOCALAPPDATA% \ConnectedDevicesPlatform folder or more commonly familiar as: C:\users\username\appdata\local\ConnectedDevicesPlatform

This folder has a file named 'CDPGlobalSettings.cdp' which is essentially a .json file

CDPGlobalSettings	
Tag	Value
<object>	
AFSEnvironment	0
AFSUrl	https://activity.windows.com
ActivityStoreInfo	
AfcDefaultUser	undefined
<object> AfcPrivacySettings	
ActivityFeed	0
CloudSync	0
PublishUserActivity	0
UploadUserActivity	1
AfsPostInitializeSyncWaitMs	10000
AfsSyncFrequencyMs	3600000
Authentication.Environment	0
BluetoothTransportEnabled	true
CcsApiVersion	/api/v1
CcsDefaultServerName	romeccs.microsoft.com
CcsPollingEnabled	false
CcsPollingInterval	0
CcsSeenRequestIds	
CcsSeenRequestIdsLastUpdate...	0000-00-00T00:00:00.000
Cloud.SessionIdleTimeoutInter...	3600
CloudTransportEnabled	true
CustomAuthClid	

and includes information for the service including the store location for the database that holds the windows timeline entries for the current user.

Tag	Value
<object>	
AFSEnvironment	0
AFSUrl	https://activity.windows.com
ActivityStoreInfo	
<element #0>	
active	true
activityStoreId	288AC2E2-C67D-6FAB-153F-7AE...
stableUserId	4a7cc0f3b728d240

The folder name may include one or more of:

- A Local account starting with an L.UserName, and/or
- folder(s) which appear to be name with random numbers but are the cid's of Microsoft accounts (e.g. Outlook.com, Live.com, Hotmail.com or MSN.com domains).

Name	Date modified
ActivitiesCache.db	24-May-18 10:55
ActivitiesCache.db-shm	24-May-18 10:54
ActivitiesCache.db-wal	24-May-18 11:00

If a user logs in to Windows with a MS account, then the respective dB is populated.

Information as to which online account has this specific ID can be found at:

NTUSER.DAT -> '[Software\Microsoft\IdentityCRL\UserExtendedProperties](#)'

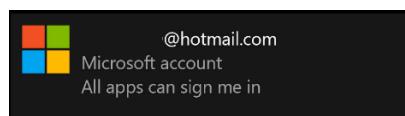
Under each online account (email) the 'cid' Value contains the respective value for each Microsoft Account & folder.

If a user logs in with a local account, then the "L.UserName" database is populated.

A Microsoft Exchange (Office365 etc.) account starts with 'AAD' and is in the form of AAD.[sid]. AAD denotes a business account (*as in Azure Active Directory*) and information associated with this account's [sid] can be found in NTUSER.dat at ->
'[Software\Microsoft\MSOIdentityCRL\UserExtendedProperties](#)' and/or
'[Software\Microsoft\OneDrive\Accounts\Business1](#)'

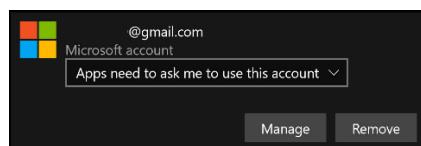
However, there are two cases when, even if a user continues to use a local account to sign in Windows 10, the activities can still be synchronized across devices:

1. The User has set up a Microsoft Account at "Settings-> Accounts->Email&app accounts" and set it up so that all apps can sign in with this account:



In this case the "L.UserName" folder is removed, and the active folder is the folder that corresponds to this account. All activities are synched across devices.

2. The User has set up a Microsoft Account at "Settings-> Accounts->Email&app accounts" and set it up so that Apps need permission to use this account:



In this case, the local activities are stored in the "L.Username" folder, but any activity on other devices is stored in the respective folder that corresponds to this MS account used on these devices.

The timeline settings ('Settings -> Privacy -> Activity History') enable or not activity to be tracked in the '[ActivitiesCache.db](#)'. In order for the activity to be shared across devices (*via the cloud*) you need to login with a Microsoft account, enable Windows Hello (i.e. create pin and/or biometric/photo login).

<https://support.microsoft.com/en-us/help/4026102/windows-10-about-sync-settings>

To enable cross-device experiences, your app users must login with either a [Microsoft Account](#) or an [Azure Active Directory](#) account.

https://github.com/Microsoft/project-rome/blob/master/cross-device_app_configuration.md



https://github.com/microsoftgraph/microsoft-graph-docs/blob/master/api-reference/beta/resources/intune_deviceconfig_windows10generalconfiguration.md

A list of settings that can be roamed or backed up (synced) across devices can be found here:
<https://docs.microsoft.com/en-us/azure/active-directory/active-directory-windows-enterprise-state-roaming-windows-settings-reference>

The 'AfcPrivacySettings' in 'CDPGlobalSettings.cdp' depicts the relative settings in Windows 10 'Settings → Privacy- >Activity History', for example:

All settings ON:



CDPGlobalSettings	
Tag	Value
> ActivityStoreInfo	
└ AfcDefaultUser	undefined
└ AfcPrivacySettings	
└ ActivityFeed	0
└ CloudSync	0
└ PublishUserActivity	0
└ UploadUserActivity	0

Upload to cloud off



AfcDefaultUser	undefined
AfcPrivacySettings	
ActivityFeed	0
CloudSync	0
PublishUserActivity	0
UploadUserActivity	1

Publish off, cloud off

Activity history

Jump back into what you were doing with apps, docs, or other activities, either on your PC or your phone.

Let Windows collect my activities from this PC

Let Windows sync my activities from this PC to the cloud

AfcPrivacySettings	
ActivityFeed	0
CloudSync	0
PublishUserActivity	1
UploadUserActivity	1

The same settings can also be configured through Group Policy at 'Computer Configuration -> Administrative Templates -> System -> OS Policies'

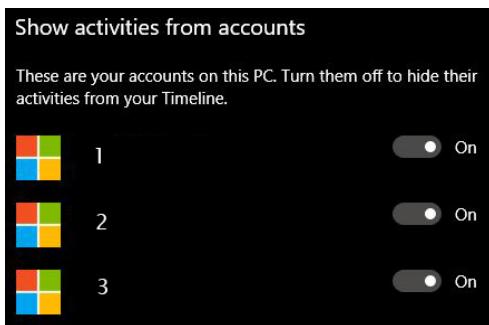
OS Policies

Setting

- Enables Activity Feed
- Allow publishing of User Activities
- Allow upload of User Activities

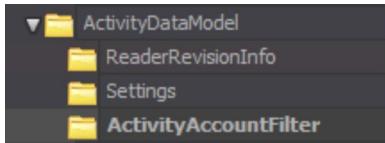
Opening the respective folder we can see the '**ActivitiesCache.db**' (*and possibly also the ActivitiesCache.wal and ActivitiesCache.db-sdb files*) which is the storage of all timeline activity for the current user. When a user switches from a Local to a MS account, the 'L.UserName' is deleted and a new one is created (if it didn't exist before) under the respective MS account folder. Any previous entries in the 'L.UserName''s '**ActivitiesCache.db**' now appear in the new '**ActivitiesCache.db**' linked to the MS account, and sync to the cloud. Switching from a MS to a local account, the MS account related db is still available and holds timeline data.

When a user logs in Windows with a MS account, and the relevant check boxes in 'Settings -> Privacy- >Activity History' are set to On,



any activity created with his other MS account(s), is updated in the '**ActivitiesCache.db**' in the respective '%LOCALAPPDATA% \ConnectedDevicesPlatform\ SubFolder, and can be seen in his timeline view .

Any -excluded from sync- accounts (*above setting set to off*) can be seen at the registry at:
NTUSER.dat
Software\Microsoft\Windows\CurrentVersion\ActivityDataModel\ActivityAccountFilter



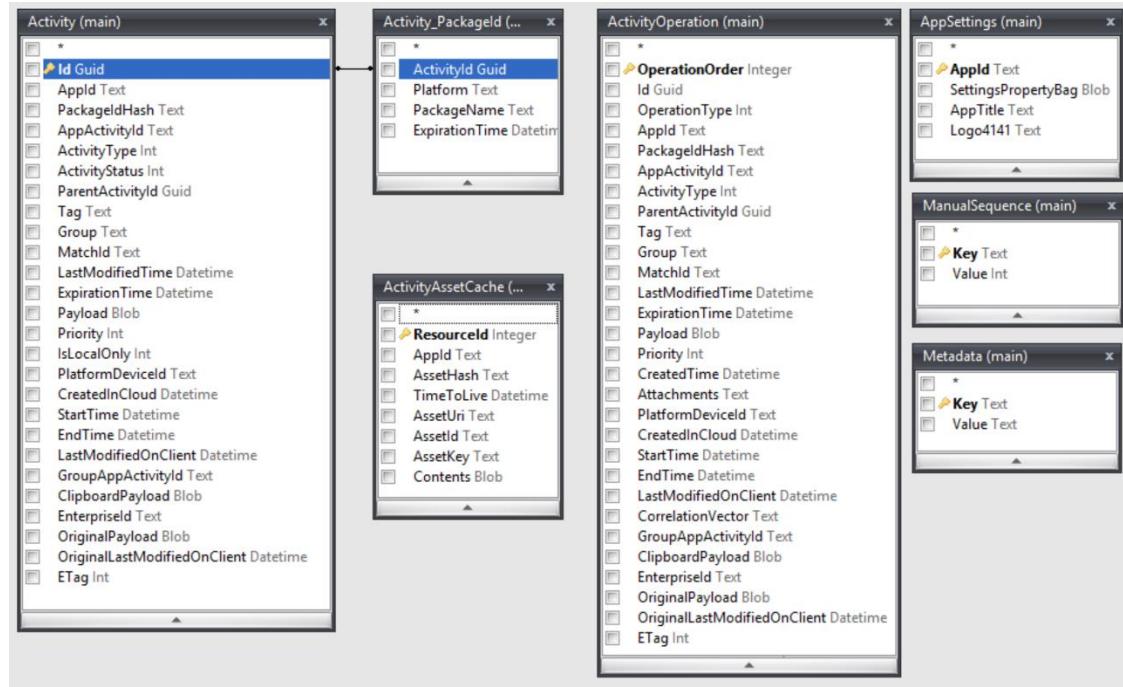
The synchronized entries in a MS account related dB will show the '**Device ID**' of the machine the entry originated from. The Device Name and Model of the originating machine can be seen in the NTUSER.dat's '**Software\Microsoft\Windows\CurrentVersion\TaskFlow\DeviceCache**':

Device ID
qcz3R4I70roQ43jHkxaTglaAC2eQunmzl6lKwNi+KOc=
qcz3R4I70roQ43jHkxaTglaAC2eQunmzl6lKwNi+KOc=
qcz3R4I70roQ43jHkxaTglaAC2eQunmzl6lKwNi+KOc=

Even if a user logs in with a local account, when a MS Account was previously used for login, the entries of '**ActivitiesCache.db**' of that account will also appear in the user's timeline (for the previous 7 days). So it may be prudent to examine all '**ActivitiesCache.db**' databases in a system.

Following is the analysis of the new 'ActivitiesCache.db' file, using [DB Browser for SQLite](#), [SQLite Expert Professional](#), [Notepad++](#) and [JSON Viewer](#).

The database has the following structure (7 tables + the master table):



The 'Metadata' table includes the date/time the database was created by the system:

The 'Metadata' table contains the following data:

	Key	Value
1	CurrentSettings	{"ActivityTypes": [4,3,2,1,0,5,13,6], "Environment": "prod"}
2	DatabaseInstanceId	3428
3	DatabaseInstanceIdUpdateTime	2018-05-24T08:16:50.365Z

Note: see page 2 above for the differences with the old database

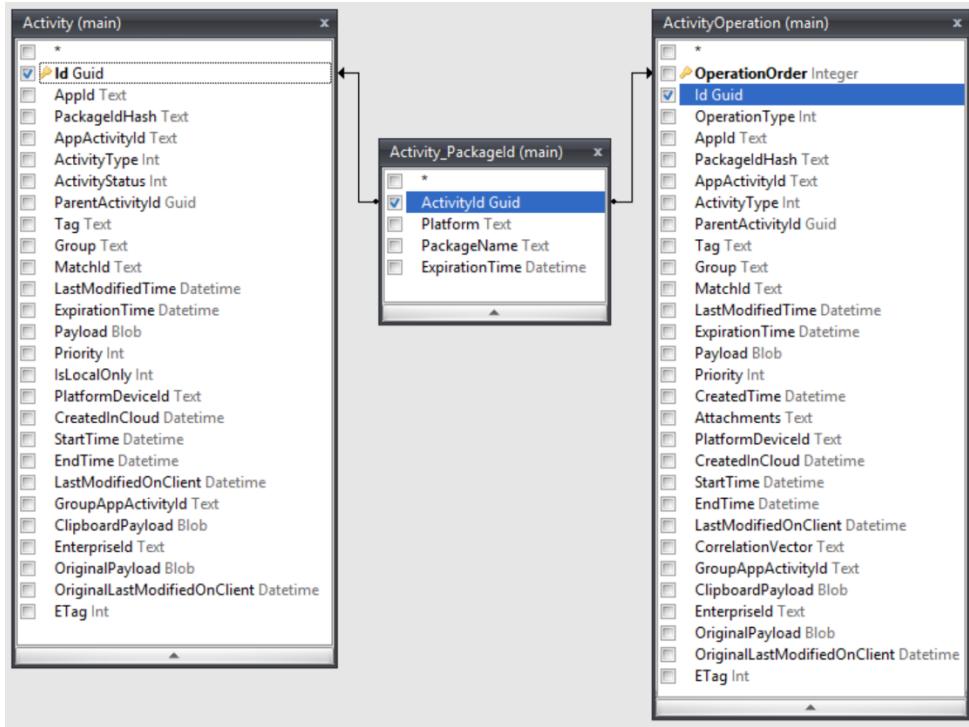
The 'ManualSequence' table shows the last Activity (ETAG) recorded in the database:

The 'ManualSequence' table contains the following data:

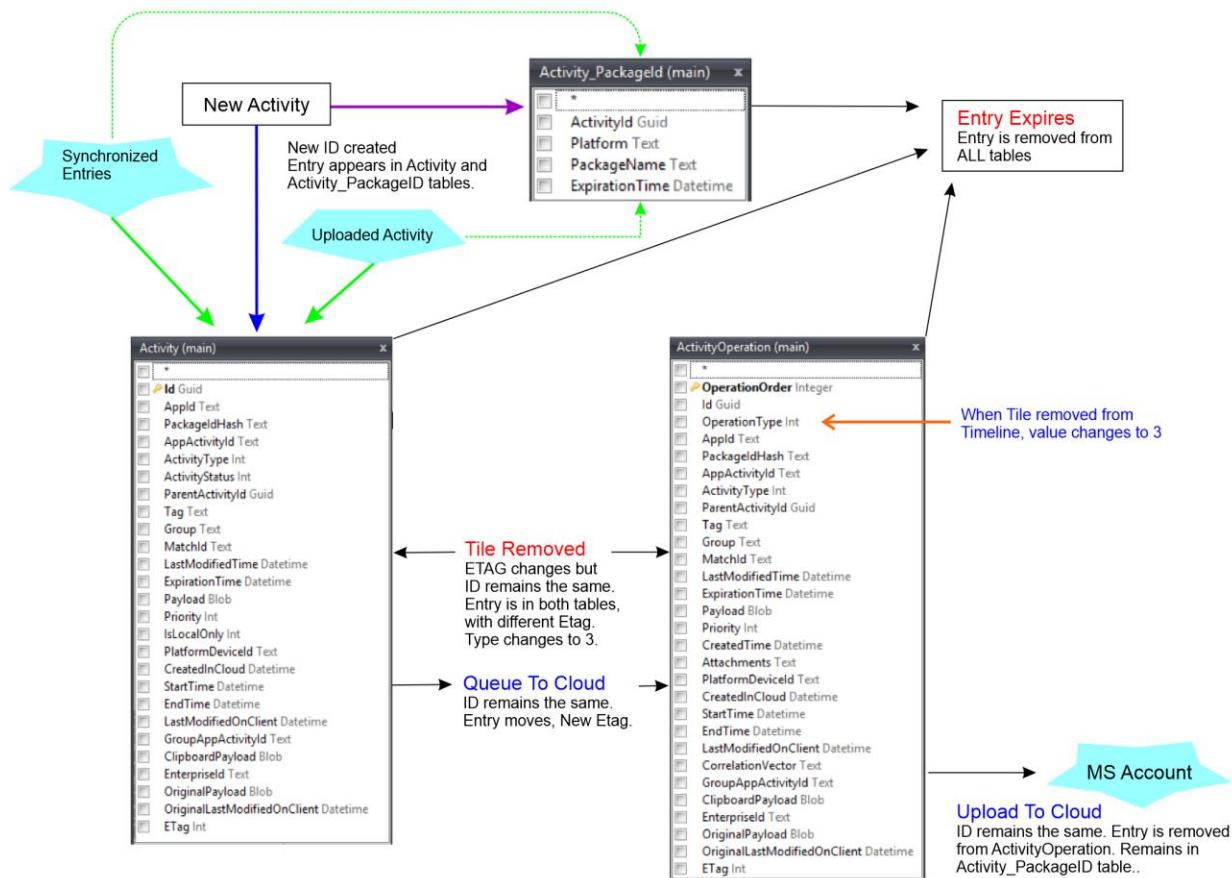
	Key	Value
1	Activity	839

From the 7 tables above, only three hold information of user activities:

'Activity', 'ActivityOperation' and 'Activity_PackageID'



The Windows Timeline process looks like this:



Any new application execution or focus change, triggers an entry to the '[Activity](#)' table, and relevant entries in the '[Activity_PackageID](#)' table.

Note: *It should be noted that any new entries/transactions are written first to the write ahead log ('ActivitiesCache.wal')*

Name	Date modified	Type	Size
ActivitiesCache.db	27-May-18 5:20 pm	DB File	4.196 KB
ActivitiesCache.db-shm	27-May-18 4:07 pm	DB-SHM File	32 KB
ActivitiesCache.db-wal	27-May-18 5:22 pm	DB-WAL File	528 KB

and the database is updated when the wal file reaches 1000 pages and commits the transactions. The default settings of the database show that each page is 4096 bytes long:

Auto Vacuum	None
Automatic Index	<input checked="" type="checkbox"/>
Checkpoint Full FSYNC	<input type="checkbox"/>
Foreign Keys	<input checked="" type="checkbox"/>
Full FSYNC	<input type="checkbox"/>
Ignore Check Constraints	<input type="checkbox"/>
Journal Mode	WAL
Journal Size Limit	-1
Locking Mode	Normal
Max Page Count	1073741823
Page Size	4096
Recursive Triggers	<input type="checkbox"/>
Secure Delete	<input type="checkbox"/>
Synchronous	Full
Temp Store	Default
User Version	18
WAL Auto Checkpoint	1000

The query below (*which is in essence two united queries*) extracts all possible useful information from this database in a readable form. This query will not work properly in any SQLite browser that does not include support for the [json1 extension](#).

```

1  SELECT -- This is the ActivityOperation Table Query
2    ActivityOperation.ETag as 'Etag',
3    json_extract(ActivityOperation.Payload, '$.appDisplayName') as 'Program Name',
4    case
5      when json_extract(ActivityOperation.AppId, '$[0].application') = '308046B0AF4A39CB'
6        then 'Firefox-308046B0AF4A39CB'
7      when json_extract(ActivityOperation.AppId, '$[1].application') = '308046B0AF4A39CB'
8        then 'Firefox-308046B0AF4A39CB'
9      when length (json_extract(ActivityOperation.AppId, '$[1].application')) > 17
10        and length (json_extract(ActivityOperation.AppId, '$[1].application')) < 22
11        then
12          replace(replace(replace(replace(
13            json_extract(ActivityOperation.AppId, '$[0].application'),
14            '([' || '6D8809377-6AF0-444B-8957-A3773F02200E' || ')', '*ProgramFiles (x64)'),
15            '([' || '7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E' || ')', '*ProgramFiles (x32)'),
16            '([' || '1AC14E77-02E7-4E5D-B744-2EB1AE5198B7' || ')', '*System'),
17            '([' || 'F38BF404-1043-42F2-9305-67DE0B28FC23' || ')', '*Windows'),
18            '([' || 'D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27' || ')', '*System32')
19        else replace(replace(replace(replace(json_extract(ActivityOperation.AppId,
20          '$[1].application'),
21            '([' || '6D8809377-6AF0-444B-8957-A3773F02200E' || ')', '*ProgramFiles (x64)'),
22            '([' || '7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E' || ')', '*ProgramFiles (x32)'),
23            '([' || '1AC14E77-02E7-4E5D-B744-2EB1AE5198B7' || ')', '*System'),
24            '([' || 'F38BF404-1043-42F2-9305-67DE0B28FC23' || ')', '*Windows'),
25            '([' || 'D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27' || ')', '*System32')
26      end as 'Application',
27      json_extract(ActivityOperation.Payload, '$.displayText')|| ' (' ||json_extract(ActivityOperation.Payload, '$.description')|| ')' as 'File/title/path
28      opened',
29      case when json_extract(ActivityOperation.Payload, '$.shellContentDescription') like '%FileShellLink%'
30        then json_extract(ActivityOperation.Payload, '$.shellContentDescription.FileShellLink')
31      else json_extract(ActivityOperation.Payload, '$.type')|| '-' ||json_extract(ActivityOperation.Payload,'$.userTimezone')
32      end as 'Payload/Timezone',
33      case ActivityOperation.ActivityType
34        when 5 then 'Open App/File/Page' when 6 then 'App In Use/Focus'
35        else 'Unknown yet'
36      end as 'Activity_type',
37      case json_extract(ActivityOperation.AppId, '$[0].platform')
38        when 'afs_crossplatform' then 'Yes'
39        when 'host' then
40          (case json_extract(ActivityOperation.AppId, '$[1].platform')
41            when 'afs_crossplatform' then 'Yes' else null end)
42        else null
43      end as 'Synced',
44      case
45        when json_extract(ActivityOperation.AppId, '$[0].platform') = 'afs_crossplatform'
46          then json_extract(ActivityOperation.AppId, '$[1].platform')
47        else json_extract(ActivityOperation.AppId, '$[0].platform')
48      end as 'Platform',
49      case ActivityOperation.OperationType
50        when 1 then 'Active' when 2 then 'Updated' when 3 then 'Deleted' when 4 then 'Ignored'
51      end as 'TileStatus',

```

```

51    case
52        when ActivityOperation.Id in
53            (select Activity.Id from Activity where Activity.Id = ActivityOperation.Id)
54            then 'Removed'
55        end as 'WasRemoved',
56    case
57        when ActivityOperation.Id
58            in(select Activity.Id from Activity where Activity.Id = ActivityOperation.Id)
59            then null else 'In Queue'
60        end as 'UploadQueue',
61        coalesce(json_extract(ActivityOperation.Payload, '$.activationUri'),json_extract(ActivityOperation.Payload, '$.reportingApp')) as 'App/Uri',
62        ActivityOperation.Priority as 'Priority',
63        time(json_extract(ActivityOperation.Payload, '$.activeDurationSeconds'),'unixepoch') as 'Active Duration',
64    case
65        when cast((ActivityOperation.EndTime - ActivityOperation.StartTime) as integer) < 0 then '-'
66        else time(cast((ActivityOperation.EndTime - ActivityOperation.StartTime) as integer),'unixepoch')
67    end as 'Calculated Duration',
68    datetime(ActivityOperation.StartTime, 'unixepoch', 'localtime') as 'StartTime',
69    datetime(ActivityOperation.LastModifiedTime, 'unixepoch', 'localtime') as 'LastModified',
70    case
71        when ActivityOperation.OriginalLastModifiedOnClient > 0
72            then datetime(ActivityOperation.OriginalLastModifiedOnClient, 'unixepoch', 'localtime')
73            else ' '
74        end as 'LastModifiedOnClient',
75    case
76        when ActivityOperation.EndTime > 0
77            then datetime(ActivityOperation.EndTime, 'unixepoch', 'localtime')
78            else null
79        end as 'EndTime',
80    case
81        when ActivityOperation.CreatedInCloud > 0
82            then datetime(ActivityOperation.CreatedInCloud, 'unixepoch', 'localtime')
83            else null
84        end as 'CreatedInCloud',
85        cast((ActivityOperation.ExpirationTime - ActivityOperation.LastModifiedTime)
86            as integer) / '86400' as 'Expires In days',
87        datetime(Activity.PackageId.ExpirationTime, 'unixepoch', 'localtime') as 'Expiration on PackageID',
88        datetime(ActivityOperation.ExpirationTime, 'unixepoch', 'localtime') as 'Expiration',
89        ActivityOperation.PlatformDeviceId as 'Device ID',
90        ActivityOperation.PackageIdHash as 'Application Hash',
91        '{' || substr(hex(Activity_PackageId.ActivityId), 1, 8) || '-' ||
92            substr(hex(Activity_PackageId.ActivityId), 9, 4) || '-' ||
93            substr(hex(Activity_PackageId.ActivityId), 13, 4) || '-' ||
94            substr(hex(Activity_PackageId.ActivityId), 17, 4) || '-' ||
95            substr(hex(Activity_PackageId.ActivityId), 21, 12) || '}' as 'ID',
96        json_extract(ActivityOperation.OriginalPayload, '$.appDisplayName') as 'Original Displayed Name',
97        json_extract(ActivityOperation.OriginalPayload, '$.displayText') as 'Original File/title opened',
98        json_extract(ActivityOperation.OriginalPayload, '$.description') as 'Original Full Path /Url',
99        coalesce(json_extract(ActivityOperation.OriginalPayload, '$.activationUri'),json_extract(ActivityOperation.OriginalPayload, '$.reportingApp')) as
100        'Original_App/Uri',
101        time(json_extract(ActivityOperation.OriginalPayload, '$.activeDurationSeconds'),'unixepoch') as 'Orig.Duration'

```

```

101
102    from Activity_PackageId
103    join ActivityOperation on Activity_PackageId.ActivityId = ActivityOperation.Id
104    where Activity_PackageId.Platform = json_extract(ActivityOperation.AppId, '$[0].platform')
105        and Activity_PackageId.ActivityId = ActivityOperation.Id
106
107 union -- Join Activity & ActivityOperation Queries to get results from both Tables
108
109 select -- This the Activity Table Query
110     Activity.ETag as 'Etag',
111     json_extract(Activity.Payload, '$.appDisplayName') AS 'Program Name',
112     case
113         when json_extract(Activity.AppId, '$[0].application') = '308046B0AF4A39CB'
114             then 'Firefox-308046B0AF4A39CB'
115         when json_extract(Activity.AppId, '$[1].application') = '308046B0AF4A39CB'
116             then 'Firefox-308046B0AF4A39CB'
117         when length (json_extract(Activity.AppId, '$[0].application')) > 17 and
118             length(json_extract(Activity.AppId, '$[0].application')) < 22
119             then replace(replace(replace(json_extract(Activity.AppId, '$[1].application'),
120                 '|||'6D809377-6AF0-444B-8957-A3773F02200E'|||'), '*ProgramFiles (x64)'),
121                 '|||'7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E'|||'), '*ProgramFiles (x32)'),
122                 '|||'1AC14E77-02E7-4E5D-B744-2EB1AE5198B7'|||'), '*System'),
123                 '|||'F38BF404-1D43-42F2-9305-67DE0B28FC23'|||'), 'Windows'),
124                 '|||'D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27'|||'), 'System32')
125         when json_extract(Activity.AppId, '$[0].application') = '308046B0AF4A39CB'
126             then 'Firefox-308046B0AF4A39CB'
127         else replace(replace(replace(replace(
128             (json_extract(Activity.AppId, '$[0].application),
129                 '|||'6D809377-6AF0-444B-8957-A3773F02200E'|||'), '*ProgramFiles (x64)'),
130                 '|||'7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E'|||'), '*ProgramFiles (x32)'),
131                 '|||'1AC14E77-02E7-4E5D-B744-2EB1AE5198B7'|||'), '*System'),
132                 '|||'F38BF404-1D43-42F2-9305-67DE0B28FC23'|||'), 'Windows'),
133                 '|||'D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27'|||'), 'System32')
134     end as 'Application',
135     json_extract(Activity.Payload, '$.displayText')|| ' (' ||json_extract(Activity.Payload, '$.description')||')' as 'File/title/path opened',
136     case when json_extract(Activity.Payload, '$.shellContentDescription') like '%FileShellLink%'
137         then json_extract(Activity.Payload, '$.shellContentDescription.FileShellLink')
138         else json_extract(Activity.Payload, '$.type')||' - ' ||json_extract(Activity.Payload,'$.userTimezone')
139     end as 'Payload/Timezone',
140     case Activity.ActivityType
141         when 5 then 'Open App/File/Page' when 6 then 'App In Use/Focus'
142         else 'Unknown yet'
143     end as 'Activity_type',
144     case json_extract(Activity.AppId, '$[0].platform')
145         when 'afs_crossplatform' then 'Yes'
146         when 'host' then (case json_extract(Activity.AppId, '$[1].platform')
147             when 'afs_crossplatform' then 'Yes' else null end) else null
148     end as 'Synced',
149     case
150         when json_extract(Activity.AppId, '$[0].platform') = 'afs_crossplatform'

```

```

151     then json_extract(Activity.AppId, '$[1].platform')
152     else json_extract(Activity.AppId, '$[0].platform')
153   end as 'Platform',
154   case Activity.ActivityStatus
155     when 1 then 'Active' when 2 then 'Updated' when 3 then 'Deleted' when 4 then 'Ignored'
156     end as 'TitleStatus',
157     null as 'WasRemoved',
158     'No' as 'UploadQueue',
159     coalesce(json_extract(Activity.Payload, '$.activationUri'),json_extract(Activity.Payload, '$.reportingApp')) as 'App/Uri',
160     Activity.Priority as 'Priority',
161     time(json_extract(Activity.Payload, '$.activeDurationSeconds'),'unixepoch') as 'Active Duration',
162   case
163     when cast ((Activity.EndTime - Activity.StartTime) as integer) < 0 then '-'
164     else time(cast((Activity.EndTime - Activity.StartTime) as integer),'unixepoch')
165   end as 'Calculated Duration',
166   datetime(Activity.StartTime, 'unixepoch', 'localtime') as 'StartTime',
167   datetime(Activity.LastModifiedTime, 'unixepoch', 'localtime') as 'LastModified',
168   case
169     when Activity.OriginalLastModifiedOnClient > 0
170       THEN datetime(Activity.OriginalLastModifiedOnClient, 'unixepoch', 'localtime')
171       ELSE '-'
172   end as 'LastModifiedOnClient',
173   case
174     when Activity.EndTime > 0
175       then datetime(Activity.EndTime, 'unixepoch', 'localtime')
176     else "-"
177   end as 'EndTime',
178   case
179     when Activity.CreatedInCloud > 0
180       then datetime(Activity.CreatedInCloud, 'unixepoch', 'localtime')
181     else "-"
182   end as 'CreatedInCloud',
183   cast((Activity.ExpirationTime - Activity.LastModifiedTime) as integer) / '86400' as 'Expires In days',
184   datetime(Activity.PackageId.ExpirationTime, 'unixepoch', 'localtime') as 'Expiration on PackageID',
185   datetime(Activity.ExpirationTime, 'unixepoch', 'localtime') as 'Expiration',
186   Activity.PlatformDeviceId as 'Device ID',
187   Activity.PackageIdHash as 'Application_Hash',
188   '{' || substr(hex(Activity.PackageId.ActivityId), 1, 8) || '-' ||
189   substr(hex(Activity.PackageId.ActivityId), 9, 4) || '-' ||
190   substr(hex(Activity.PackageId.ActivityId), 13, 4) || '-' ||
191   substr(hex(Activity.PackageId.ActivityId), 17, 4) || '-' ||
192   substr(hex(Activity.PackageId.ActivityId), 21, 12) || '}' as 'ID',
193   json_extract(Activity.OriginalPayload, '$.appDisplayName') as 'Original Program Name',
194   json_extract(Activity.OriginalPayload, '$.displayText') as 'Original File/title opened',
195   json_extract(Activity.OriginalPayload, '$.description') as 'Original Full Path /Url',
196   coalesce(json_extract(Activity.OriginalPayload, '$.activationUri'),json_extract(Activity.OriginalPayload, '$.reportingApp')) as 'Original_App/Uri',
197   time(json_extract(Activity.OriginalPayload, '$.activeDurationSeconds'),'unixepoch') as 'Orig.Duration'
198
199 from Activity_PackageId
200 join Activity on Activity_PackageId.ActivityId = Activity.Id
201 where Activity_PackageId.Platform = json_extract(Activity.AppId, '$[0].platform')
202   and Activity_PackageId.ActivityId = Activity.Id
203
204 order by Etag desc; -- Edit this Line to change the sorting

```

An updated version of this query can be found online at:
<https://kacos2000.github.io/WindowsTimeline/>

The above query pulls information from the ‘Activity’, ‘ActivityOperation’ and ‘Activity_PackageId’ tables. The ‘Activity_PackageId’ table is more or less a ‘cache’ of the unique IDs of each application execution and its expiration time.

Some field explanations provided by [Microsoft](#):

Name	Type	Description
status	EnumType	Set by the server. A status code used to identify valid objects. Values: active, updated, deleted, ignored.
userTimezone	String	Optional. The timezone in which the user's device used to generate the activity was located at activity creation time. Values supplied as Olson IDs in order to support cross-platform representation.
createdDateTime	DateTimeOffset	Set by the server. DateTime in UTC when the object was created on the server.
lastModifiedDateTime	DateTimeOffset	Set by the server. DateTime in UTC when the object was modified on the server.
id	String	Required. Client-set GUID for the historyItem object.
startedDateTime	DateTimeOffset	Required. UTC DateTime when the historyItem (activity session) was started. Required for timeline history.
lastActiveDateTime	DateTimeOffset	Optional. UTC DateTime when the historyItem (activity session) was last understood as active or finished - if null, historyItem status should be Ongoing.
expirationDateTime	DateTimeOffset	Optional. UTC DateTime when the historyItem will undergo hard-delete. Can be set by the client.
activeDurationSeconds	int	Optional. The duration of active user engagement. If not supplied, this is calculated from the startedDateTime and lastActiveDateTime .

The **Activity**' table's '**ID**' (*Primary key*) field is a unique value:

[Activity]([Id] GUID PRIMARY KEY NOT NULL

It is the unique ID of each entry/user activity. The same ID can be seen more than 2 times in the **Activity_PackageID** table depending on the '**Platform**' field which has the following possible values (*Note: most of project Rome is still undocumented, so not all entries can be defined*):

- “**Host**” (*is listed as many times as the number of entries in the Activity table*),
- “**packageid**”**, (*is listed as many times as the number of entries and includes the executable and its path for all **x_exe** & **windows_win32** apps, as seen in the **Activity_PackageID**’s **PackageName** field*)
- “**x_exe_path**” for Programs called from a specific path (e.g. Standalone executables)

** as seen below:

ETag	ID	Activity_PackageId.PackageName	Activity_json_Packageid
41723	3254D1271207F6C50CED...	*System\cmd.exe	*System\cmd.exe
41722	2FB37475807AA6C1D3C9...	*System\snippingtool.exe	*System\SnippingTool.exe
41721	3A99ABF8D3BA3ABE6B31...	microsoft.office.desktop_8wekyb3d8bbwe!powerpoint	NULL
41720	935EB39CF7CE0C170659...	*ProgramFiles (x64)\isoft technologies\active@ disk editor...	*ProgramFiles (x64)\Soft Technologies\Active@ Disk Edi...
41719	2922B987D85B0B6FA4ED...	*ProgramFiles (x64)\tracker software\pdf viewer\pdfxcview...	*ProgramFiles (x64)\Tracker Software\PDF Viewer\PDFXC...
41718	F13910B8FE851B13E14D...	microsoft.windowscalculator_8wekyb3d8bbwe!app	Microsoft.WindowsCalculator_8wekyb3d8bbwe!App
41717	FF1241976D7A557AE1E...	microsoft.windowscalculator_8wekyb3d8bbwe!app	Microsoft.WindowsCalculator_8wekyb3d8bbwe!App
41715	D222BC19A52035C3398E...	*System\snippingtool.exe	*System\SnippingTool.exe
41713	5A61C68F4DE1C99F14A4...	*System\snippingtool.exe	*System\SnippingTool.exe
41707	587C1298F7A2D926DBD...	microsoft.windows.explorer	Microsoft.Windows.Explorer
41706	68547C5BDBFD068D2F3F...	microsoft.windows.explorer	Microsoft.Windows.Explorer
41704	A7B05CED895ABC4DCCF...	*System\notepad.exe	*System\notepad.exe

By running the following query:

```

select
Activity.ETag,
hex(Activity.PackageId.ActivityId) as 'ID',
replace(replace(replace(replace(replace(
    Activity_PackageId.PackageName,
    lower('308046B0AF4A39CB'), 'Mozilla Firefox'),
    '{'||lower('6D809377-6AF0-444B-8957-A3773F02200E'||')', '*ProgramFiles (x64)'),
    '{'||lower('7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E'||')', '*ProgramFiles (x32)'),
    '{'||lower('1AC14E77-02E7-4E5D-B744-2EB1AE5198B7'||')', 'System'),
    '{'||lower('F38BF404-1D43-42F2-9305-67DE0B28FC23'||')', 'Windows'),
    '{'||lower('D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27'||')', 'System32') as
'Activity_PackageId.PackageName',
case
    when json_extract(Activity.AppId, '$[0].platform') like '%packageid%' then
replace(replace(replace(replace(
        replace(
            (json_extract(Activity.AppId, '$[0].application'),
            '308046B0AF4A39CB', 'Mozilla Firefox'),
            '{'||'6D809377-6AF0-444B-8957-A3773F02200E'||')', '*ProgramFiles (x64)'),
            '{'||'7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E'||')', '*ProgramFiles (x32)'),
            '{'||'1AC14E77-02E7-4E5D-B744-2EB1AE5198B7'||')', 'System'),
            '{'||'F38BF404-1D43-42F2-9305-67DE0B28FC23'||')', 'Windows'),
            '{'||'D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27'||')', 'System32')
    when json_extract(Activity.AppId, '$[1].platform') like '%packageid%' then
replace(replace(replace(replace(
        replace(
            (json_extract(Activity.AppId, '$[1].application'),
            '308046B0AF4A39CB', 'Mozilla Firefox'),
            '{'||'6D809377-6AF0-444B-8957-A3773F02200E'||')', '*ProgramFiles (x64)'),
            '{'||'7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E'||')', '*ProgramFiles (x32)'),
            '{'||'1AC14E77-02E7-4E5D-B744-2EB1AE5198B7'||')', 'System'),
            '{'||'F38BF404-1D43-42F2-9305-67DE0B28FC23'||')', 'Windows'),
            '{'||'D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27'||')', 'System32')
    when json_extract(Activity.AppId, '$[2].platform') like '%packageid%' then
replace(replace(replace(replace(
        replace(
            (json_extract(Activity.AppId, '$[2].application'),
            '308046B0AF4A39CB', 'Mozilla Firefox'),
            '{'||'6D809377-6AF0-444B-8957-A3773F02200E'||')', '*ProgramFiles (x64)'),
            '{'||'7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E'||')', '*ProgramFiles (x32)'),
            '{'||'1AC14E77-02E7-4E5D-B744-2EB1AE5198B7'||')', 'System'),
            '{'||'F38BF404-1D43-42F2-9305-67DE0B28FC23'||')', 'Windows'),
            '{'||'D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27'||')', 'System32')
    when json_extract(Activity.AppId, '$[3].platform') like '%packageid%' then
replace(replace(replace(replace(
        replace(
            (json_extract(Activity.AppId, '$[3].application'),
            '308046B0AF4A39CB', 'Mozilla Firefox'),
            '{'||'6D809377-6AF0-444B-8957-A3773F02200E'||')', '*ProgramFiles (x64)'),
            '{'||'7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E'||')', '*ProgramFiles (x32)'),
            '{'||'1AC14E77-02E7-4E5D-B744-2EB1AE5198B7'||')', 'System'),
            '{'||'F38BF404-1D43-42F2-9305-67DE0B28FC23'||')', 'Windows'),
            '{'||'D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27'||')', 'System32')
    when json_extract(Activity.AppId, '$[4].platform') like '%packageid%' then
replace(replace(replace(replace(
        replace(
            (json_extract(Activity.AppId, '$[4].application'),
            '308046B0AF4A39CB', 'Mozilla Firefox'),
            '{'||'6D809377-6AF0-444B-8957-A3773F02200E'||')', '*ProgramFiles (x64)'),
            '{'||'7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E'||')', '*ProgramFiles (x32)'),
            '{'||'1AC14E77-02E7-4E5D-B744-2EB1AE5198B7'||')', 'System'),
            '{'||'F38BF404-1D43-42F2-9305-67DE0B28FC23'||')', 'Windows'),
            '{'||'D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27'||')', 'System32')

```

```

when json_extract(Activity.AppId, '$[5].platform') like '%packageid%' then
replace(replace(replace(replace(replace
    (json_extract(Activity.AppId, '$[5].application'),
    '308046B0AF4A39CB', 'Mozilla Firefox'),
    '{' || '6D809377-6AF0-444B-8957-A3773F02200E'||'|}', '**ProgramFiles (x64)'),
    '{' || '7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E'||'|}', '**ProgramFiles (x32)'),
    '{' || '1AC14E77-02E7-4E5D-B744-2EB1AE519887'||'|}', '*System'),
    '{' || 'F38BF404-1D43-42F2-9305-67DE0B28FC23'||'|}', '**Windows'),
    '{' || 'D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27'||'|}', '**System32')
when json_extract(Activity.AppId, '$[6].platform') like '%packageid%' then
replace(replace(replace(replace(replace
    (json_extract(Activity.AppId, '$[6].application'),
    '308046B0AF4A39CB', 'Mozilla Firefox'),
    '{' || '6D809377-6AF0-444B-8957-A3773F02200E'||'|}', '**ProgramFiles (x64)'),
    '{' || '7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E'||'|}', '**ProgramFiles (x32)'),
    '{' || '1AC14E77-02E7-4E5D-B744-2EB1AE519887'||'|}', '*System'),
    '{' || 'F38BF404-1D43-42F2-9305-67DE0B28FC23'||'|}', '**Windows'),
    '{' || 'D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27'||'|}', '**System32')
when json_extract(Activity.AppId, '$[7].platform') like '%packageid%' then
replace(replace(replace(replace(replace
    (json_extract(Activity.AppId, '$[7].application'),
    '308046B0AF4A39CB', 'Mozilla Firefox'),
    '{' || '6D809377-6AF0-444B-8957-A3773F02200E'||'|}', '**ProgramFiles (x64)'),
    '{' || '7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E'||'|}', '**ProgramFiles (x32)'),
    '{' || '1AC14E77-02E7-4E5D-B744-2EB1AE519887'||'|}', '*System'),
    '{' || 'F38BF404-1D43-42F2-9305-67DE0B28FC23'||'|}', '**Windows'),
    '{' || 'D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27'||'|}', '**System32')
when json_extract(Activity.AppId, '$[8].platform') like '%packageid%' then
replace(replace(replace(replace(replace
    (json_extract(Activity.AppId, '$[8].application'),
    '308046B0AF4A39CB', 'Mozilla Firefox'),
    '{' || '6D809377-6AF0-444B-8957-A3773F02200E'||'|}', '**ProgramFiles (x64)'),
    '{' || '7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E'||'|}', '**ProgramFiles (x32)'),
    '{' || '1AC14E77-02E7-4E5D-B744-2EB1AE519887'||'|}', '*System'),
    '{' || 'F38BF404-1D43-42F2-9305-67DE0B28FC23'||'|}', '**Windows'),
    '{' || 'D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27'||'|}', '**System32') end as 'Activity_json_Packageid'

from Activity_PackageId
join Activity on Activity.Id      = Activity_PackageId.ActivityId
where Activity_PackageId.ActivityId = Activity.Id and Activity_PackageId.Platform = 'packageid'
Order by etag desc

```

plus one or more of the following ([platform-specific application IDs](#)):

- “[windows_win32](#)” for Installed Software (non UWP) Applications
- “[windows_universal](#)” for Windows [UWP](#) Apps
- “[msa](#)”,
- “[ios](#)”,
- “[android](#)”, and
- “[web](#)”

As seen by this [MS example](#):

Example:

```
{"platform": "windows_universal", "application": "Microsoft.Contoso_8wekyb3d8bbw"},  
{"platform": "windows_win32", "application": "DefaultBrowser_NOPUBLISHERID!Microsoft.Contoso.Default"},  
{"platform": "android", "application": "com.example.myapp"},  
{"platform": "ios", "application": "com.example.myapp"},  
{"platform": "web", "application": "https://contoso.com"},  
{"platform": "web", "application": "https://chat.contoso.com"},  
{"platform": "msa", "application": "0000000603E0BF"},  
 {"platform": "msa", "application": "48932b46-98b1-4020-9be4-cc7a65643c9e"},  
]
```

If the Sync activity is enabled (user logs in with a MS Account) then another entry may also be seen: “[afs_crossplatform](#)” which shows a [cloud sync](#) enabled activity, and is an additional entry to the same ones seen when using a local account.

When using a MS account:

5251	afs_crossplatform	6082723978844556501	1529741627
5252	windows_win32	{7c5a40ef-a0fb-4bfc-874a-c0f2e0b9fa8e}\foobar2000\f...	1529741627
5253	packageid	{7c5a40ef-a0fb-4bfc-874a-c0f2e0b9fa8e}\foobar2000\f...	1529741627
5254	host		1529741627

When using a Local account there is NO 'afs_crossplatform' entry:

79	windows_win32	microsoft.window...	1529742843
80	packageid	microsoft.window...	1529742843
81	host		1529742843

As can also be seen below (*these are for a single ID of a Microsoft Word entry*):

ActivityId	Platform	PackageName	ExpirationTime
CA06838559F87AE076659BF18C593CF9	host	word.activity.windows.com	2018-06-29 16:39:49
CA06838559F87AE076659BF18C593CF9	afs_crossplatform	16883319693718197224	2018-06-29 16:39:49
CA06838559F87AE076659BF18C593CF9	windows_universal	microsoft.office.word_8wekyb3d8bbwe	2018-06-29 16:39:49
CA06838559F87AE076659BF18C593CF9	windows_universal	microsoft.office.desktop_8wekyb3d8bbwe	2018-06-29 16:39:49
CA06838559F87AE076659BF18C593CF9	android	com.microsoft.office.word	2018-06-29 16:39:49
CA06838559F87AE076659BF18C593CF9	ios	com.microsoft.office.word	2018-06-29 16:39:49
CA06838559F87AE076659BF18C593CF9	windows_win32	microsoft.office.winword.exe.15	2018-06-29 16:39:49
CA06838559F87AE076659BF18C593CF9	windows_win32	{7c5a40ef-a0fb-4bfc-874a-c0f2e0b9fa8e}\microsoft office\office15\winword.exe	2018-06-29 16:39:49
CA06838559F87AE076659BF18C593CF9	windows_win32	{6d809377-6af0-444b-8957-a3773f02200e}\microsoft office\office15\winword.exe	2018-06-29 16:39:49
CA06838559F87AE076659BF18C593CF9	windows_win32	{7c5a40ef-a0fb-4bfc-874a-c0f2e0b9fa8e}\microsoft office\office14\winword.exe	2018-06-29 16:39:49
CA06838559F87AE076659BF18C593CF9	windows_win32	{6d809377-6af0-444b-8957-a3773f02200e}\microsoft office\office14\winword.exe	2018-06-29 16:39:49
CA06838559F87AE076659BF18C593CF9	msa	0000000000003edf6	2018-06-29 16:39:49

The above can be seen by using this query:

```
Select
hex(Activity_PackageId.ActivityId) as 'ActivityId',
    Activity_PackageId.Platform ,
    Activity_PackageId.PackageName ,
    datetime (Activity_PackageId.ExpirationTime , 'unixepoch') as 'ExpirationTime'
FROM Activity_PackageId
join Activity on Activity_PackageId.ActivityId = Activity.Id
order by Activity.ActivityId
```

An example of the entry of Microsoft YourPhone's app:

```
[{"application":"mmxsdktest.azurewebsites.net","platform":"host"}, {"application":"16815419600048394801","platform":"afs_crossplatform"}, {"application":"000000004018945C","platform":"msa"}, {"application":"","platform":"packageId"}, {"application":"","platform":"alternateId"}, {"application":"","platform":"windows_universal"}]
```

The afs_crossplatform field has been observed to contain a numeric representation of the application used (?), with the exception of MS Office applications and MS Edge:

Activity_json_afs_crossplatform
9553497022742795366
9616461484989533944
9726032056939584244
972940661883734646
9751790172484571661
9792636623328491085
edge.activity.windows.com
excel.activity.windows.com
powerpoint.activity.windows.com
word.activity.windows.com

As seen by running the query below:

```
select
case
when json_extract(Activity.AppId, '$[0].platform') like '%afs_crossplatform%' then json_extract(Activity.AppId, '$[0].application')
when json_extract(Activity.AppId, '$[1].platform') like '%afs_crossplatform%' then json_extract(Activity.AppId, '$[0].application')
when json_extract(Activity.AppId, '$[2].platform') like '%afs_crossplatform%' then json_extract(Activity.AppId, '$[0].application')
when json_extract(Activity.AppId, '$[3].platform') like '%afs_crossplatform%' then json_extract(Activity.AppId, '$[0].application')
when json_extract(Activity.AppId, '$[4].platform') like '%afs_crossplatform%' then json_extract(Activity.AppId, '$[0].application')
when json_extract(Activity.AppId, '$[5].platform') like '%afs_crossplatform%' then json_extract(Activity.AppId, '$[0].application')
when json_extract(Activity.AppId, '$[6].platform') like '%afs_crossplatform%' then json_extract(Activity.AppId, '$[0].application')
when json_extract(Activity.AppId, '$[7].platform') like '%afs_crossplatform%' then json_extract(Activity.AppId, '$[0].application')
when json_extract(Activity.AppId, '$[8].platform') like '%afs_crossplatform%' then json_extract(Activity.AppId, '$[0].application')
end as 'Activity_json_afs_crossplatform'
from Activity
group by Activity_json_afs_crossplatform |
```

The 'Activity_PackageId' table seemingly has more information than the 'Activity' table. That is not true though, as the 'Activity' table's 'AppID' field holds this information and a lot more. The 'Activity' and 'ActivityOperation' table 'AppID' fields are in fact Binary Large Objects (BLOB) in 'json array' format e.g.:

'AppID' entry for an Installed Application in Windows:

<array>	
<element #0>	{7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E}\TeamViewer\TeamViewer.exe windows_win32
<element #1>	{7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E}\TeamViewer\TeamViewer.exe packageId
<element #2>	alternateId
<element #3>	windows_universal

Note: Notice that the path shown above is in the form of a GUID. It is a KNOWNFOLDERID used by windows for standard known folders like 'program files'. In the json file above, this is the GUID for ProgramFilesX86 (C:\Program Files (x86)\) folder. There are various lists of CLSIDs in the internet like [https://msdn.microsoft.com/en-us/library/windows/desktop/dd378457\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd378457(v=vs.85).aspx)

'AppID' entry for a standalone executable:

Tag	Value
<array>	
<element #0>	
application	D:\Forensic Tools\XMLView\xMLView.exe
platform	x_exe_path
<element #1>	
application	D:\Forensic Tools\XMLView\xMLView.exe
platform	packageId
<element #2>	
application	
platform	alternateId
<element #3>	
application	
platform	windows_universal

and 'AppID' entry for a Windows UWP app

Tag	Value
<array>	
<element #0>	
application	Microsoft.MicrosoftStickyNotes_8wekyb3d8bbwe!App
platform	windows_universal
<element #1>	
application	Microsoft.MicrosoftStickyNotes_8wekyb3d8bbwe!App
platform	packageId
<element #2>	
application	
platform	alternateId

Since this is a json array, we can do a query to see what values it holds in the 'application' and 'platform' fields. The `$[x].` depicts the number of the element in the JSON array, and 'platform' or 'application' of the respective field name:

```
-- List all entries in the AppID json array

select
hex(Activity.ID) as 'ID',
json_extract(Activity.AppId, '$[0].platform') as 'p0' ,
json_extract(Activity.AppId, '$[0].application') as '0' ,
json_extract(Activity.AppId, '$[1].platform') as 'p1' ,
json_extract(Activity.AppId, '$[1].application') as '1' ,
json_extract(Activity.AppId, '$[2].platform') as 'p2' ,
json_extract(Activity.AppId, '$[2].application') as '2' ,
json_extract(Activity.AppId, '$[3].platform') as 'p3' ,
json_extract(Activity.AppId, '$[3].application') as '3' ,
json_extract(Activity.AppId, '$[4].platform') as 'p4' ,
json_extract(Activity.AppId, '$[4].application') as '4' ,
json_extract(Activity.AppId, '$[5].platform') as 'p5' ,
json_extract(Activity.AppId, '$[5].application') as '5' ,
json_extract(Activity.AppId, '$[6].platform') as 'p6' ,
```

```

    json_extract(Activity.AppId, '$[6].application') as '6' ,
    json_extract(Activity.AppId, '$[7].platform') as 'p7' ,
    json_extract(Activity.AppId, '$[7].application') as '7' ,
    json_extract(Activity.AppId, '$[8].platform') as 'p8' ,
    json_extract(Activity.AppId, '$[8].application') as '8'

```

```

from Activity
order by ID asc

```

	A0	P0	A1	P1	A2	P2	A3	P3
19	{7C5A40EF-A0FB-4BFC-874A-C0F2E089FA8E}\Acronis\TrueImageHome\TrueImage.exe	x_exe_path	{7C5A40EF-A0FB-4BFC-874A-C0F2...	packageId	alternated		windows_universal	
20	Microsoft.Office.OUTLOOK.EXE.15		windows_win32	Microsoft.Office.OUTLOOK.EXE.15	packageId	alternated	windows_universal	
21	{6D809377-6AF0-444B-8957-A3773F02200E}\VOW Software\plist Editor Pro\plistEditor.exe	windows_win32	{6D809377-6AF0-444B-8957-A377...	packageId	alternated		windows_universal	
22	{7C5A40EF-A0FB-4BFC-874A-C0F2E089FA8E}\Acronis\TrueImageHome\TrueImage.exe	x_exe_path	{7C5A40EF-A0FB-4BFC-874A-C0F2...	packageId	alternated		windows_universal	
23	{6D809377-6AF0-444B-8957-A3773F02200E}\SQLite Expert\Professional 5\SQLiteExpertP...	windows_win32	{6D809377-6AF0-444B-8957-A377...	packageId	alternated		windows_universal	
24	Microsoft.Office.OUTLOOK.EXE.15	windows_win32	Microsoft.Office.OUTLOOK.EXE.15	packageId	alternated		windows_universal	

Or

	2	p3	3	p4	4	p5	5	p6	6	p7	7	p8	8
1	osoftEdge_8wekyb3d8bbwe!MicrosoftEdge	msa	f44b1140-bc5e-48c6-8dc0-5cf5a53c0e34	msa	00000004C1BC462	android.com.microsoft.emmx.daily	android.com.microsoft.emmx.selfhost	android.com.microsoft.emmx	ios	com.microsoft.emmx.daily-df			
2	osoftEdge_8wekyb3d8bbwe!MicrosoftEdge	msa	f44b1140-bc5e-48c6-8dc0-5cf5a53c0e34	msa	00000004C1BC462	android.com.microsoft.emmx.daily	android.com.microsoft.emmx.selfhost	android.com.microsoft.emmx	ios	com.microsoft.emmx.daily-df			
3	osoftEdge_8wekyb3d8bbwe!MicrosoftEdge	msa	f44b1140-bc5e-48c6-8dc0-5cf5a53c0e34	msa	00000004C1BC462	android.com.microsoft.emmx.daily	android.com.microsoft.emmx.selfhost	android.com.microsoft.emmx	ios	com.microsoft.emmx.daily-df			
4	osoftEdge_8wekyb3d8bbwe!MicrosoftEdge	msa	f44b1140-bc5e-48c6-8dc0-5cf5a53c0e34	msa	00000004C1BC462	android.com.microsoft.emmx.daily	android.com.microsoft.emmx.selfhost	android.com.microsoft.emmx	ios	com.microsoft.emmx.daily-df			
5	osoftEdge_8wekyb3d8bbwe!MicrosoftEdge	msa	f44b1140-bc5e-48c6-8dc0-5cf5a53c0e34	msa	00000004C1BC462	android.com.microsoft.emmx.daily	android.com.microsoft.emmx.selfhost	android.com.microsoft.emmx	ios	com.microsoft.emmx.daily-df			

Similarly, the '[Activity](#)' and the '[ActivityOperation](#)' table 'payload' fields (blob) are also in json format with different entries related to the different platform type as follows:

For "[x_exe_path](#)" , "[windows_win32](#)" and "[windows_universal](#)" it doesn't include duration [when an app is first executed](#):

Tag	Value
object	
displayText	JSONView.exe
activationUri	ms-shellactivity:
appDisplayName	JSONView.exe
backgroundColor	black

but includes it in subsequent entries:

Tag	Value
object	
type	UserEngaged
reportingApp	ShellActivityMonitor
activeDurationSeconds	16
shellContentDescription	
MergedGap	600
userTimezone	Europe/Bucharest

Having included my own duration calculation (calculated by subtracting the 'StartTime' from the 'EndTime'), the duration is sometimes different than the one included in the json blob. A longer calculated duration could indicate that the app is in focus but idle i.e. the user does not interact with it.

When "windows_universal" or "windows_win32" applications open a file, the 'payload' field includes quite a bit of information.

"[windows_universal](#)" (In this case MS Edge - which uses the [Adaptive Card framework](#) - bing search for CDPTraces.log) can be seen below:

Tag	Value
<object>	
displayText	CDPTraces.log - Bing
activationUri	microsoft-edge: https://www.bing.com/search?q=CDPTraces.log&form=WNSGPH&qs=S...
appDisplayName	Microsoft Edge
description	https://www.bing.com/search?q=CDPTraces.log&form=WNSGPH&qs=S...
backgroundColor	#FF0078D7
adaptiveContent	
\$schema	http://adaptivecards.io/schemas/adaptive-card.json
type	AdaptiveCard
body	
<element #0>	Container
type	TextBlock
text	CDPTraces.log - Bing
weight	bolder
size	large
wrap	true
maxLines	3
<element #1>	TextBlock
type	https://www.bing.com/search?q=CDPTraces.log&form=WNSGPH&qs=S...
text	normal
size	true
wrap	3
maxLines	https://www.bing.com/search?q=CDPTraces.log&form=WNSGPH&qs=S...
contentUri	
attribution	
iconUri	https://www.bing.com/sa/img/bing_p_rr_teal_min.ico
alternateText	bing.com
attributionDisplayText	bing.com

And '[windows_win32](#)' here:

Tag	Value
<object>	
displayText	Photos.sqlite_embedded_table-ZGENERICALBUM_rowid-42_column-ZCLC
activationUri	ms-shellactivity:
appDisplayName	plist Editor Pro
description	D:\Costas\Desktop\temp\iOS 11iPhone Image\Photos.sqlite_embedded_
backgroundColor	black
contentUri	file:%7BB4BFCC3A-DB2C-424C-B029-7FE99A87C641%7D/.temp\iOS%2
shellContentDescription	
FileShellLink	MBAAAAEFCAAAAAAAAADAAAAAAAY0gAAAAgAAAAAsAnjYCVpPdALw5Im

The 'ExpirationTime' fields in both **Activity** and **Activity_PackageId** tables are the same for each 'ID' and 'ActivityId' respectively (Unique field entry GUID).

All Dates/Times are stored by the database in UTC. To get the local times, we can adjust the query by using, 'localtime' e.g. :

```
datetime(Activity.StartTime , 'unixepoch', 'localtime'),
```

The 'ExpirationTime' is exactly 2592000 seconds or 30 Days from the relevant entry's 'LastModified' time, and is easily seen by using the following:

```
Select
(Activity.ExpirationTime - Activity.LastModifiedTime) as 'Expires in'
From Activity
```

Sorting the query by the 'ETAG' field gives a historical list (sort of a timeline) according to the Events recorded in the db, but is not the same as if sorted by Start, End or Expiry datetimes. The ETAG can change for a particular entry's GUID depending on the current action the database has recorded for that ID (e.g. user removed Tile from timeline).

An example that an ID can have multiple ETAGs associated with it, is shown below:

ID	Program Name	Nr_of_ETAGs_per_ID
1 14A18F08264CCBEEE1F87294159CC174	Microsoft.MicrosoftEdge_8wekyb3d8bbwe!MicrosoftEdge	12
2 E410C45C9351D584F5FF8771FB79EBE0	Microsoft.MicrosoftEdge_8wekyb3d8bbwe!MicrosoftEdge	12
3 3850384E31582BB4D7444A87238EBE71	Microsoft.MicrosoftEdge_8wekyb3d8bbwe!MicrosoftEdge	11
4 7FBCA86B1284531AB13714378DE462BF	Microsoft.MicrosoftEdge_8wekyb3d8bbwe!MicrosoftEdge	11
5 AF83A1D79E8018D88E34C6DA0367D8D2	Microsoft.MicrosoftEdge_8wekyb3d8bbwe!MicrosoftEdge	11
6 005A5B3C02AF18D3D743923C12A6A1A4	{1AC14E77-02E7-4E5D-B744-2EB1AE5198B7}\SnippingTool.exe	4
7 00AEEED61A3676A61C4683F4ADF956DC	{6D809377-6AF0-444B-8957-A3773F02200E}\Windows NT\Accessories\wordpad...	4
8 0197B6BDC361072CAE085484AF9E9D13	{6D809377-6AF0-444B-8957-A3773F02200E}\Windows NT\Accessories\wordpad...	4
9 019F2517AFD0B40206B668B4B979C5F3	Chrome	4
10 026D393D2A3C851046DD8296FFFFA0C1	Chrome	4

by running this query against a MS account's 'ActivitiesCache.db':

```

1 select
2 hex(Activity.id) as 'ID',
3 json_extract(Activity.AppId, '$[2].application') as 'Program Name',
4 count(activity.etag) as 'Nr_of_ETAGs_per_ID'
5 from Activity
6 join Activity_PackageId on activity.id = Activity_PackageId.ActivityId
7 group by ID
8 order by Nr_of_ETAGs_per_ID desc

```

The 'PlatformDeviceID' seen in the 'Activity' & 'ActivityOperations' tables:

A screenshot of a database management system interface. On the left, there is a table with a single column labeled 'PlatformDeviceId'. The first few rows show values like '7it7kvkbMZ9R20MhUL/UCkDTVs44...' repeated several times. On the right, there is a text area containing the following SQL command:

```

506 PRAGMA ignore_check_constraint;
507 PRAGMA journal_mode;
508 PRAGMA journal_size_limit;

```

Below this, there is a button labeled 'Edit Database Cell' and a dropdown menu set to 'Text'. At the bottom of the text area, the value '7it7kvkbMZ9R20MhUL/UCkDTVs44...' is highlighted.

can be found in the user's NTUSER.dat at

"Software\Microsoft\Windows\CurrentVersion\TaskFlow\DeviceCache\".

As seen below:

A screenshot of a registry editor. On the left, there is a tree view of registry keys under 'Key name'. One key is expanded, showing several subkeys. On the right, there is a table with columns 'Value Name', 'Value Type', and 'Data'. The data shows the following entries:

Value Name	Value Type	Data
DeviceName	RegSz	ALIEN
DeviceType	RegDword	15
DeviceMake	RegSz	Alienware
DeviceModel	RegSz	Alienware 15 R3

But it is not fixed - it looks encoded with the [QuickXorHash Algorithm](#), similar to the 'FileShellLink' in the 'Payload' field blob - and it changes occasionally,

A screenshot of a database management system interface. On the left, there is a table with a single column labeled 'Device ID'. The first few rows show values like 'TDdtGmqdscQk7BxppyGhb2yzThGYW8Ldm8tsq+GJ9n4=' repeated several times. On the right, there is a text area containing the following SQL command:

```

190 TDdtGmqdscQk7BxppyGhb2yzThGYW8Ldm8tsq+GJ9n4=
191 TDdtGmqdscQk7BxppyGhb2yzThGYW8Ldm8tsq+GJ9n4=
192 TDdtGmqdscQk7BxppyGhb2yzThGYW8Ldm8tsq+GJ9n4=
193 TDdtGmqdscQk7BxppyGhb2yzThGYW8Ldm8tsq+GJ9n4=
194 Yd9BtyKlyWtQS5jTPjr+weOyQ/ceqpRRyL6aX0XAiro=
195 Yd9BtyKlyWtQS5jTPjr+weOyQ/ceqpRRyL6aX0XAiro=
196 Yd9BtyKlyWtQS5jTPjr+weOyQ/ceqpRRyL6aX0XAiro=

```

still depicting the same Machine (as seen at the relevant registry entries). I have not yet found the pattern as to why or when it changes when a user logs in with a local account.

According to [Microsoft](#) “..Note that the hardware hash also contains details about when it was generated, so it will change each time it is generated..” which makes sense.

It is used to show the Device* origin of a Tile in the Timeline view, such as:



**(if no device name is seen in one or more tiles in the user's Timeline view, then the activity is local)*

The device name seen above is derived from the “DeviceModel” entry in the registry for the specific DeviceID. That information originates from the OEM information value in ‘SystemProductName’ at “HKLM\SYSTEM\ControlSet001\Control\SystemInformation” which is usually filled by the device manufacturer, or is blank when the computer is user assembled.

The ‘AppActivityId’ field has 3 possible observed values:

- ECB32AF3-1440-4086-94E3-5311F97F89C4 which looks like a GUID but having checked 4 different computers of different users except my own ones, this GUID remains the same in all of them.
- An alphanumeric identifier (Hash?) following the above GUID, which is specific to a file opened, or
- A URL opened by MS Edge (on MS Account related ‘ActivitiesCache.db’ databases).

This can be seen by running the following:

```
select
case when Activity.AppActivityId not like '%-%-%-%-%%' then Activity.AppActivityId
when Activity.AppActivityId not like '%-%-%-%-%' then Activity.AppActivityId
else substr(Activity.AppActivityId , 38)
end as 'DeviceID',
datetime(Activity.StartTime, 'unixepoch', 'localtime') AS 'StartTime',
json_extract(Activity.Payload, '$.displayText') as 'File_Name'
from Activity
join Activity_PackageId on activity.id = Activity_PackageId.ActivityId
group by File_Name
order by StartTime asc
```

	DeviceID	StartTime	File_Name
13	ECB32AF3-1440-4086-94E3-5311F97F89C4	2018-05-01 07:20:10	7-Zip File Manager
14	ECB32AF3-1440-4086-94E3-5311F97F89C4	2018-05-01 07:24:15	SystemPropertiesAdvanced.exe
15	ECB32AF3-1440-4086-94E3-5311F97F89C4	2018-05-01 07:28:48	Opera Browser
16	ECB32AF3-1440-4086-94E3-5311F97F89C4	2018-05-01 07:35:45	ESEDatabaseView.exe
17	ECB32AF3-1440-4086-94E3-5311F97F89C4	2018-05-01 07:45:59	jre-8u171-windows-x64.exe
18	4cc59f0fe7ab8e7280b2c4228bcd76193b3e5f3f	2018-05-01 07:58:11	backup.tar
19	fc1379ad2d57897a2013f92a6e5d53e0dbd1e0f8	2018-05-01 08:01:38	tealium.db
20	b6e7e75f16f5d0fdd4260be2a2e1058bcfb619a	2018-05-01 08:05:13	flixster_dc2.db

The ‘**PackageIDHash**’ field is a unique value related to each application. It appears to be the ‘[QuickXorHash](#)’ of the executable – with this query:

```
select
    Activity.PackageIdHash as 'Hash',
    Activity_PackageId.PackageName as 'Name',
    datetime(Activity.StartTime, 'unixepoch', 'localtime') AS 'StartTime'
    from Activity
    join Activity_PackageId on activity.id = Activity_PackageId.ActivityId
group by Hash
order by StartTime desc
```

we can see that different versions of the same application have a different Hash value:

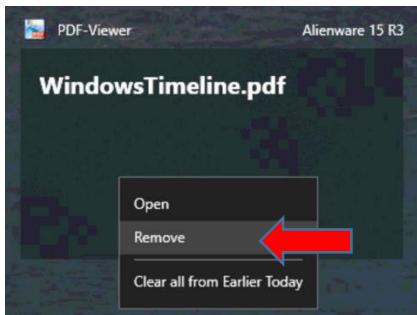
	Hash	Name	StartTime
1	LpMbiNsalvhwz9KScqy+21v9qKArm8vsmGRxfqVh8k=	d:\forensic tools\dmde-3.2.0.692-win32-gui\dmde.exe	2018-05-28 19:44:18
2	83TnIWHEKlgYDl4QLfHkrWYjNrlRwf7OP4R6Djg6TXI=	d:\forensic tools\dmde-3.0.6.648-win32-gui\dmde.exe	2018-05-28 19:44:00

The ‘[Activity](#)’ table’s ‘[ActivityStatus](#)’ and ‘[ActivityOperation](#)’ tables’s ‘[OperationType](#)’ have 3 observed values ranging from 1 to 3 and depict the related Tile Status as:

- 1 -> Active (means it is still an open application)
- 2-> Updated (means the entry is an update for a previously Active entry)
- 3-> Deleted (means the Tile is removed from the Timeline)

And there [could be a 4th entry](#) with a value of 4, meaning that the entry is ‘Ignored’.

When a Tile is removed from the Timeline by the user,



the associated ID(s) to this Tile remain the same, but the entries are copied from the ‘[Activity](#)’ to the ‘[ActivityOperation](#)’ table with a NEW ‘[Etag](#)’ and new type/status value:

```

1 select
2 activity.etag as 'Activity Etag',
3 hex(activity.id) as 'Activity ID',
4 Activity.ActivityStatus as 'Type',
5 ActivityOperation.ETag as 'ActivityOperation Etag',
6 hex(ActivityOperation.Id) as 'ActivityOperation ID',
7 ActivityOperation.OperationType as 'Type'
8 from Activity
9 join ActivityOperation on Activity.Id = ActivityOperation.Id

```

	Activity Etag	Activity ID	Type	ActivityOperation Etag	ActivityOperation ID	Type
1	19255	942964941C4A1FC75A837A1CFC8CFF92	1	20098	942964941C4A1FC75A837A1CFC8CFF92	3
2	19264	0DE4C1CFA7AD59E7B9F9A3A0509D21CA	1	20099	0DE4C1CFA7AD59E7B9F9A3A0509D21CA	3
3	19267	4B8DDEC337073252145BE8B622D2A619	1	20100	4B8DDEC337073252145BE8B622D2A619	3
4	19270	3704910DA81C809DB04126F03E300092	1	20101	3704910DA81C809DB04126F03E300092	3
5	19410	289D11B903A8BC64FE6305B96BC46750	1	20102	289D11B903A8BC64FE6305B96BC46750	3

When the user logs in with a local account, and a tile is removed from the Timeline, the associated entries for that are copied from the '[Activity](#)' to the '[ActivityOperation](#)' table with a different '[ETAG](#)' value, and stay there until expiration time.

An easy way to figure out which entries are User Removed can be derived from viewing the '[SmartLookup](#)' view (*automatically created query included in the database*) which, in essence, is the source of the actual timeline a user observes:

```

1  select
2      [O].[Id],
3      [O].[AppId],
4      [O].[PackageIdHash],
5      [O].[AppActivityId],
6      [O].[ActivityType],
7      [O].[OperationType] AS [ActivityStatus],
8      [O].[ParentActivityId],
9      [O].[Tag],
10     [O].[Group],
11     [O].[MatchId],
12     [O].[LastModifiedTime],
13     [O].[ExpirationTime],
14     [O].[Payload],
15     [O].[Priority],
16     [A].[IsLocalOnly],
17     [O].[PlatformDeviceId],
18     [A].[CreatedInCloud],
19     [O].[StartTime],
20     [O].[EndTime],
21     [O].[LastModifiedOnClient],
22     1 AS [IsInUploadQueue],
23     [O].[GroupAppActivityId],
24     [O].[ClipboardPayload],
25     [O].[EnterpriseId],
26     [O].[OriginalPayload],
27     [O].[OriginalLastModifiedOnClient],
28     [O].[ETag]
29  from  [ActivityOperation] as [O]
30  left outer join [Activity] as [A] on [O].[Id] = [A].[Id]
31 union
32 select
33     [Id],
34     [AppId],
35     [PackageIdHash],
36     [AppActivityId],
37     [ActivityType],
38     [ActivityStatus],
39     [ParentActivityId],
40     [Tag],
41     [Group],
42     [MatchId],
43     [LastModifiedTime],
44     [ExpirationTime],
45     [Payload],
46     [Priority],
47     [IsLocalOnly],
48     [PlatformDeviceId],
49     [CreatedInCloud],
50     [StartTime],
51     [EndTime],
52     [LastModifiedOnClient],
53     0 AS [IsInUploadQueue],
54     [GroupAppActivityId],
55     [ClipboardPayload],
56     [EnterpriseId],
57     [OriginalPayload],
58     [OriginalLastModifiedOnClient],
59     [ETag]
60  from  [Activity]
61  where [Id] not in (select [Id]
62      from  [ActivityOperation])

```

The key here is the line: `FROM [Activity] WHERE [Id] NOT IN (SELECT [Id] FROM [ActivityOperation])`

If we check the ‘SmartLookup’ query data view, it lists all ‘Activity’ table entries plus all ‘ActivityOperation’ table entries minus the ‘Activity’ table entries that appear in both tables with the same ID. Why? Because when the `ActivityOperation.Id` is also found in the `Activity.Id` then the entry has been (user) removed from the timeline.

What this query also does, is to mark (and list) all entries in the ‘Activity’ table with a value of 0 in the field [IsInUploadQueue], and all entries in the ‘ActivityOperations’ with a value of 1 in the [IsInUploadQueue]. What this means, is that the entries in the ‘ActivityOperations’ table are in the ‘Upload to Cloud Queue’ unless they also appear in the ‘Activity’ table.

However, when a user is logged in with a MS Account, and the user removes a tile from the Timeline, the respective entries are copied to the ‘ActivityOperation’ table with a new ‘Etag’ value and a new type/status of 3 (*Deleted*) momentarily until they are synced, and then they are

moved to the ‘Activity’ table until they expire. The type/status value 3 tells the other synchronized devices not to display this tile in their Timeline. This may be confusing when examining an offline system:

Table: ActivityOperation

	OperationOrder	Id	OperationType	AppId
	Filter	Filter	Filter	Filter
1	1	◆d◆ J ◆Z...	3	[{"application":...
2	2	◆◆3◆Y◆◆...	3	[{"application":...
3	3	K◆◆◆7 2R ...	3	[{"application":...
4	4	BLOB	3	[{"application":...
5	5	(◆ ◆ ◆d...	3	[{"application":...



The following query checks every ID on ‘Activity_PackageID’ against both ‘Activity’ and ‘ActivityOperation’ tables and provides the relevant results:

Archived:

ID not in either table.



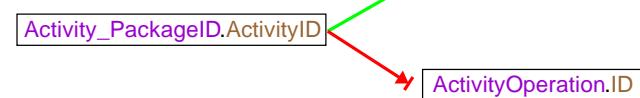
Tile Removed:

ID in both tables.



New Activity:

ID only in the ‘Activity’ table.



In Upload Queue:

ID only in ‘ActivityOperation’ table.



```

select
case
when
    Activity_PackageId.ActivityId in (select Activity.Id from activity) and
    Activity_PackageId.ActivityId in (select ActivityOperation.Id
                                    from ActivityOperation)
then 'Tile Removed - '|hex(Activity_PackageId.ActivityId)
else
    case
    when      Activity_PackageId.ActivityId not in
                (select Activity.Id from activity) and
                Activity_PackageId.ActivityId not in
                (select ActivityOperation.Id from ActivityOperation)
            then 'Is Archived - '|hex(Activity_PackageId.ActivityId)
    else
  
```

```

        case
        when Activity_PackageId.ActivityId in
        (select Activity.Id from activity) and
        Activity_PackageId.ActivityId not in
        (select ActivityOperation.Id from ActivityOperation)
            then 'New Activity - '||hex(Activity_PackageId.ActivityId)
        else
        case
            when Activity_PackageId.ActivityId
            not in (select Activity.Id from Activity)
            and Activity_PackageId.ActivityId
            in (select ActivityOperation.Id from ActivityOperation)
            then '
        In Upload Queue - '||hex(Activity_PackageId.ActivityId)
        end
        end
    end
end as 'Status_ID', -- This field includes both the Status and the unique ID of the associated activity
Activity_PackageId.PackageName as 'PackageName', -- The program/ associated with the above ID
datetimetz(Activity_PackageId.ExpirationTime, 'unixepoch', 'localtime') as 'ExpirationTime',
Activity_PackageId.Platform

from Activity_PackageId
where Activity_PackageId.Platform in ('windows_win32', 'windows_universal', 'x_exe_path')
group by Status_ID
order by ExpirationTime asc

```

We can see which ID's entry is a:

- 'New activity'
- 'Had the associated timeline tile removed'
- 'In the Upload Queue'
- Archived' (*until expiration time*)

	Status_ID	PackageName	ExpirationTime	Platform
2464	New Activity - FAC3BF8F86CC8D737E2912333C1B480B	microsoft.office.outlook.exe.15	2018-08-21 18:29:32	windows_win32
2465	New Activity - A082487B90E1920C1E794CCE485D46D6	microsoft.office.outlook.exe.15	2018-08-21 18:29:54	windows_win32
2466	Is Archived - AA02BBBF0A259DEAE764099B3E5BCB8B	(6d809377-6af0-444b-8957-a3773f02200e)\db browser for sqlite\db browser for sqlite.exe	2018-08-24 10:30:25	windows_win32
2467	Is Archived - 5D82FFF1A179FFEAC9A59CE52FC48ACA	windows.immersivecontrolpanel_cw5n1h2txyewy	2018-08-24 10:30:44	windows_universal
2468	New Activity - E4D4D3EE8FF161D4D330C2C5FAAEECF2	(6d809377-6af0-444b-8957-a3773f02200e)\db browser for sqlite\db browser for sqlite.exe	2018-08-24 10:30:44	windows_win32
2469	Is Archived - F48D10A0C52199D9D35BDB1EE68CBC4E	windows.immersivecontrolpanel_cw5n1h2txyewy	2018-09-25 23:06:50	windows_universal
2470	Is Archived - 65D2974DC6A859499B8B20FA926146BE	microsoft.office.outlook.exe.15	2018-11-21 17:29:58	windows_win32
2471	Is Archived - 732C5C66B18DBB39B19AAE2D48AA8C65	windows.immersivecontrolpanel_cw5n1h2txyewy	2018-11-21 17:29:59	windows_universal

The '**Priority**' field has these possible values (*and assumed explanations*):

- 0 None
- 1 High
- 3 Low

As indicated by [MS Graph REST API](#): "*Indicates the priority of a raw user notification. Visual notifications are sent with high priority by default. Valid values are None, High and Low.*"

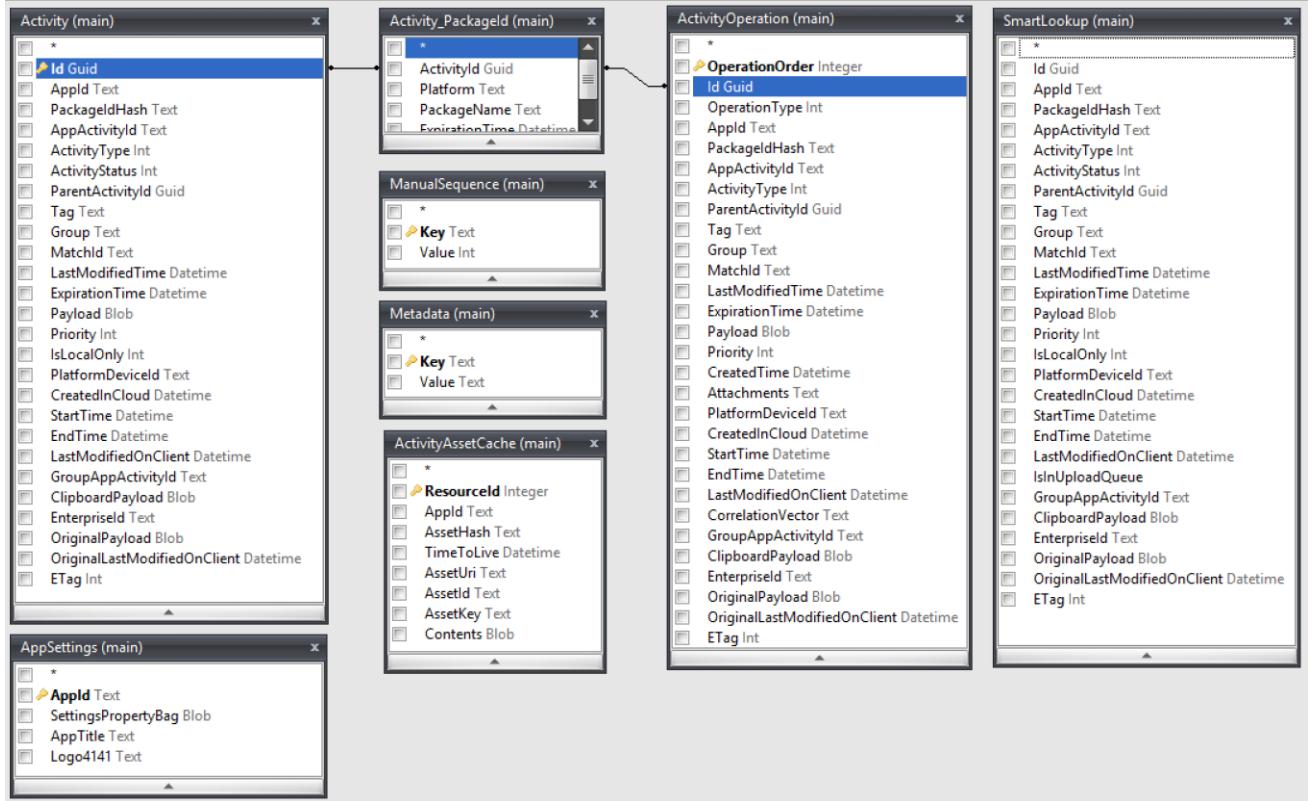
Note: The '**userTimezone**' information found in '**ActivityType**' 6 entries's '**Payload**' field is:
 "...the timezone in which the user's device used to generate the activity was located at activity creation time; values supplied as [Olson ID](#)'s in order to support cross-platform representation."

[Source](#)

Changes with Windows 10 version 1809 (*October 2018 update*)

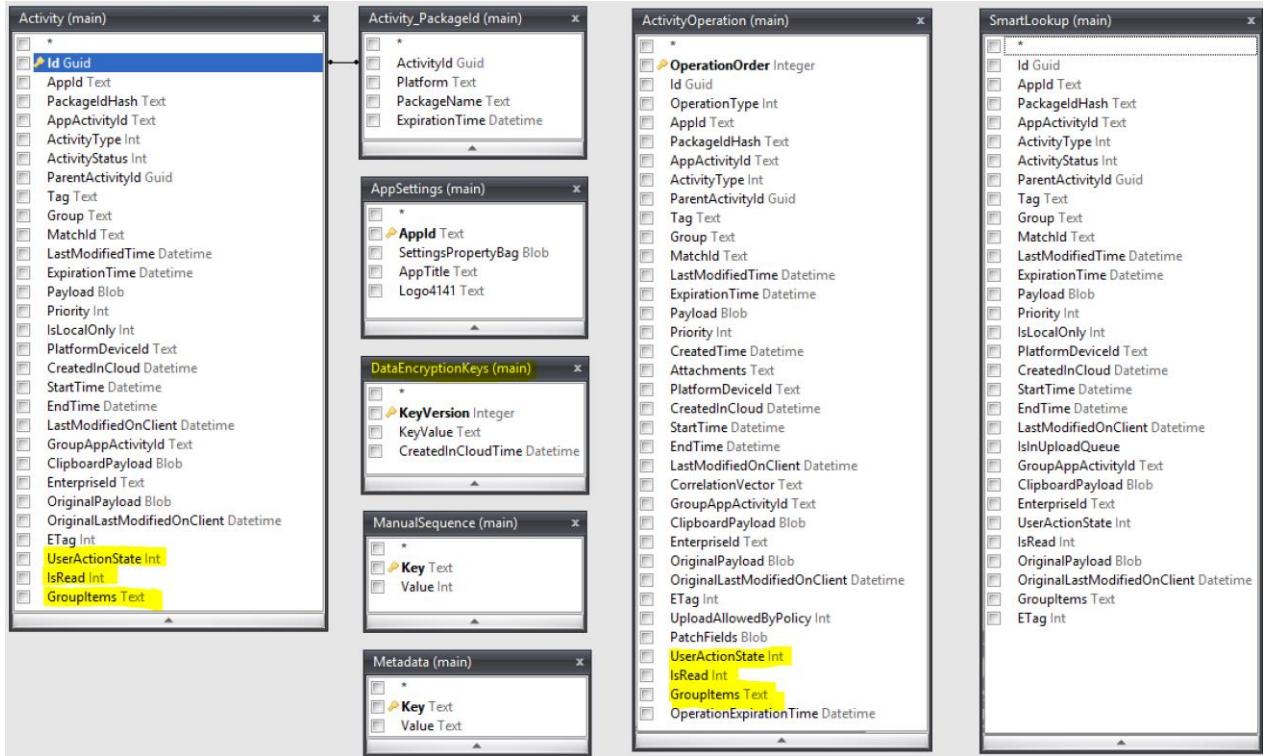
(The following is based on Win10 Insider's build 17744.rc5_release.180818-1845)

The previous (1803) was structured like this:



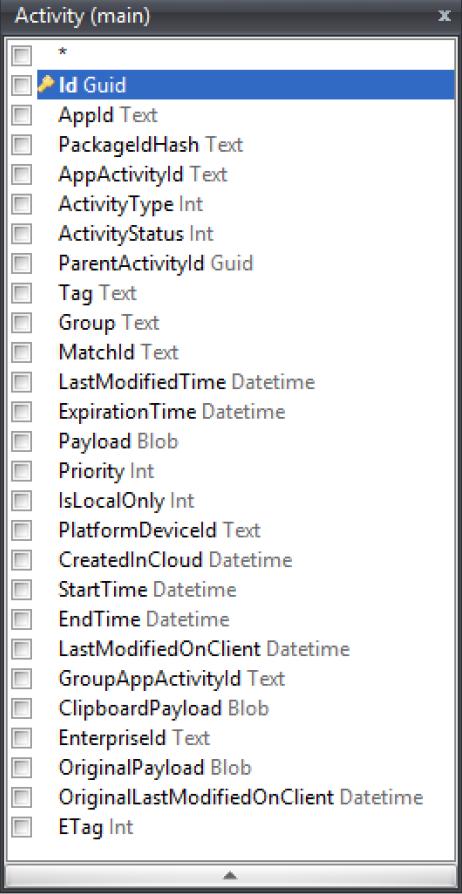
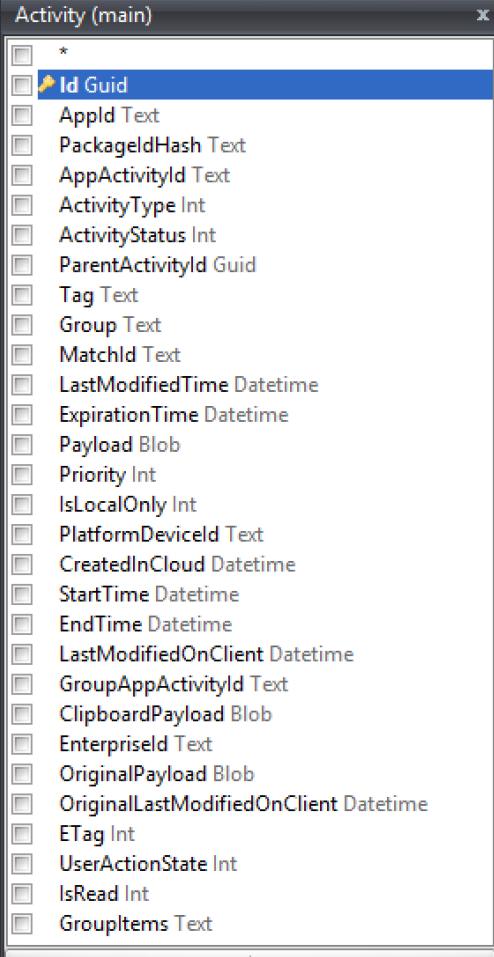
With the main tables being 'Activities' and 'ActivitiesOperation', while 'Activities_PackageID' was the link and archive of all the activity entries.

The database structure has changed a bit with the new October 2018 Windows 10 update (version 1809). A new table is introduced:



As well as new fields in both Activity & ActivityOperation tables:

Activity	ActivityOperation
	UploadAllowedByPolicy
UserActionState	UserActionState
IsRead	IsRead
GroupItems	GroupItems

Before (1803)	After (1809)
 <p>The screenshot shows the 'Activity (main)' entity editor. The 'Id' field is selected, indicated by a blue highlight bar above the field name. Other fields listed include AppId, PackageHash, AppActivityId, ActivityType, ActivityStatus, ParentActivityId, Tag, Group, MatchId, LastModifiedTime, ExpirationTime, Payload, Priority, IsLocalOnly, PlatformDeviceId, CreatedInCloud, StartTime, EndTime, LastModifiedOnClient, GroupAppActivityId, ClipboardPayload, EnterpriseId, OriginalPayload, and OriginalLastModifiedOnClient.</p>	 <p>The screenshot shows the 'Activity (main)' entity editor after version 1809. The 'Id' field is selected, indicated by a blue highlight bar above the field name. The list of fields is identical to the one shown in the 'Before' screenshot.</p>

ActivityOperation (main)	
<ul style="list-style-type: none"> <input type="checkbox"/> * <input checked="" type="checkbox"/> OperationOrder Integer <input type="checkbox"/> Id Guid <input type="checkbox"/> OperationType Int <input type="checkbox"/> AppId Text <input type="checkbox"/> PackageldHash Text <input type="checkbox"/> AppActivityId Text <input type="checkbox"/> ActivityType Int <input type="checkbox"/> ParentActivityId Guid <input type="checkbox"/> Tag Text <input type="checkbox"/> Group Text <input type="checkbox"/> MatchId Text <input type="checkbox"/> LastModifiedTime Datetime <input type="checkbox"/> ExpirationTime Datetime <input type="checkbox"/> Payload Blob <input type="checkbox"/> Priority Int <input type="checkbox"/> CreatedTime Datetime <input type="checkbox"/> Attachments Text <input type="checkbox"/> PlatformDeviceId Text <input type="checkbox"/> CreatedInCloud Datetime <input type="checkbox"/> StartTime Datetime <input type="checkbox"/> EndTime Datetime <input type="checkbox"/> LastModifiedOnClient Datetime <input type="checkbox"/> CorrelationVector Text <input type="checkbox"/> GroupAppActivityId Text <input type="checkbox"/> ClipboardPayload Blob <input type="checkbox"/> Enterpriseld Text <input type="checkbox"/> OriginalPayload Blob <input type="checkbox"/> OriginalLastModifiedOnClient Datetime <input type="checkbox"/> ETag Int 	<ul style="list-style-type: none"> <input type="checkbox"/> * <input checked="" type="checkbox"/> OperationOrder Integer <input type="checkbox"/> Id Guid <input type="checkbox"/> OperationType Int <input type="checkbox"/> AppId Text <input type="checkbox"/> PackageldHash Text <input type="checkbox"/> AppActivityId Text <input type="checkbox"/> ActivityType Int <input type="checkbox"/> ParentActivityId Guid <input type="checkbox"/> Tag Text <input type="checkbox"/> Group Text <input type="checkbox"/> MatchId Text <input type="checkbox"/> LastModifiedTime Datetime <input type="checkbox"/> ExpirationTime Datetime <input type="checkbox"/> Payload Blob <input type="checkbox"/> Priority Int <input type="checkbox"/> CreatedTime Datetime <input type="checkbox"/> Attachments Text <input type="checkbox"/> PlatformDeviceId Text <input type="checkbox"/> CreatedInCloud Datetime <input type="checkbox"/> StartTime Datetime <input type="checkbox"/> EndTime Datetime <input type="checkbox"/> LastModifiedOnClient Datetime <input type="checkbox"/> CorrelationVector Text <input type="checkbox"/> GroupAppActivityId Text <input type="checkbox"/> ClipboardPayload Blob <input type="checkbox"/> Enterpriseld Text <input type="checkbox"/> OriginalPayload Blob <input type="checkbox"/> OriginalLastModifiedOnClient Datetime <input type="checkbox"/> ETag Int <input type="checkbox"/> UploadAllowedByPolicy Int <input type="checkbox"/> PatchFields Blob <input type="checkbox"/> UserActionState Int <input type="checkbox"/> IsRead Int <input type="checkbox"/> GroupItems Text <input type="checkbox"/> OperationExpirationTime Datetime

Table: ManualSequence													
<table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Filter</td> <td>Filter</td> </tr> <tr> <td>1 Activity</td> <td>126910</td> </tr> </tbody> </table>	Key	Value	Filter	Filter	1 Activity	126910	<table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Filter</td> <td>Filter</td> </tr> <tr> <td>1 Activity</td> <td>127355</td> </tr> </tbody> </table>	Key	Value	Filter	Filter	1 Activity	127355
Key	Value												
Filter	Filter												
1 Activity	126910												
Key	Value												
Filter	Filter												
1 Activity	127355												

Table: Metadata																															
<table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Filter</td> <td>Filter</td> </tr> <tr> <td>1 CurrentSettings</td> <td>{"ActivityTypes": [4,3,2,1,0,5,13,6], "Environment": "prod"}</td> </tr> <tr> <td>2 DatabaseInstanceId</td> <td>23919</td> </tr> <tr> <td>3 DatabaseInstanceIdUpdateTime</td> <td>2018-05-10T13:58:59.758Z</td> </tr> <tr> <td>4 CurrentEtag</td> <td>bbaa7770-adbf-11e8-9117-01020305070d</td> </tr> </tbody> </table>	Key	Value	Filter	Filter	1 CurrentSettings	{"ActivityTypes": [4,3,2,1,0,5,13,6], "Environment": "prod"}	2 DatabaseInstanceId	23919	3 DatabaseInstanceIdUpdateTime	2018-05-10T13:58:59.758Z	4 CurrentEtag	bbaa7770-adbf-11e8-9117-01020305070d	<table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Filter</td> <td>Filter</td> </tr> <tr> <td>1 CurrentSettings</td> <td>{"ActivityTypes": [0,1,2,3,4,5,6,7,11,12,13,15], "Environment": "prod"}</td> </tr> <tr> <td>2 DatabaseInstanceId</td> <td>23919</td> </tr> <tr> <td>3 DatabaseInstanceIdUpdateTime</td> <td>2018-05-10T13:58:59.758Z</td> </tr> <tr> <td>4 CurrentEtag</td> <td>ec9c6840-addb-11e8-9117-01020305070d</td> </tr> <tr> <td>5 DatabaseNotificationSubscriptionInfo</td> <td>{"partialSyncToken": "", "publisherFilters": [{"activityTypes": [0,1,2,3,4,5,6,7,13,15], "application": "*"}, {"activityTypes": [1,2,3,4,5,6,7,10,13,15], "application": "*"}], "allowedSubscriptionSyncScopes": [{"allowedTypes": [0,1,2,3,4,5,6,7,10,13,15], "application": "*"}, {"allowedTypes": []}], "activityTypes": []}</td> </tr> <tr> <td>6 DatabaseActivityPolicies</td> <td></td> </tr> <tr> <td>7 PendingActivityTypes</td> <td></td> </tr> </tbody> </table>	Key	Value	Filter	Filter	1 CurrentSettings	{"ActivityTypes": [0,1,2,3,4,5,6,7,11,12,13,15], "Environment": "prod"}	2 DatabaseInstanceId	23919	3 DatabaseInstanceIdUpdateTime	2018-05-10T13:58:59.758Z	4 CurrentEtag	ec9c6840-addb-11e8-9117-01020305070d	5 DatabaseNotificationSubscriptionInfo	{"partialSyncToken": "", "publisherFilters": [{"activityTypes": [0,1,2,3,4,5,6,7,13,15], "application": "*"}, {"activityTypes": [1,2,3,4,5,6,7,10,13,15], "application": "*"}], "allowedSubscriptionSyncScopes": [{"allowedTypes": [0,1,2,3,4,5,6,7,10,13,15], "application": "*"}, {"allowedTypes": []}], "activityTypes": []}	6 DatabaseActivityPolicies		7 PendingActivityTypes	
Key	Value																														
Filter	Filter																														
1 CurrentSettings	{"ActivityTypes": [4,3,2,1,0,5,13,6], "Environment": "prod"}																														
2 DatabaseInstanceId	23919																														
3 DatabaseInstanceIdUpdateTime	2018-05-10T13:58:59.758Z																														
4 CurrentEtag	bbaa7770-adbf-11e8-9117-01020305070d																														
Key	Value																														
Filter	Filter																														
1 CurrentSettings	{"ActivityTypes": [0,1,2,3,4,5,6,7,11,12,13,15], "Environment": "prod"}																														
2 DatabaseInstanceId	23919																														
3 DatabaseInstanceIdUpdateTime	2018-05-10T13:58:59.758Z																														
4 CurrentEtag	ec9c6840-addb-11e8-9117-01020305070d																														
5 DatabaseNotificationSubscriptionInfo	{"partialSyncToken": "", "publisherFilters": [{"activityTypes": [0,1,2,3,4,5,6,7,13,15], "application": "*"}, {"activityTypes": [1,2,3,4,5,6,7,10,13,15], "application": "*"}], "allowedSubscriptionSyncScopes": [{"allowedTypes": [0,1,2,3,4,5,6,7,10,13,15], "application": "*"}, {"allowedTypes": []}], "activityTypes": []}																														
6 DatabaseActivityPolicies																															
7 PendingActivityTypes																															

The Metadata table is the one major change. The Current Settings key, as previously, is where the allowed Activity Types are listed:

```
1 CurrentSettings {"ActivityTypes": [0,1,2,3,4,5,6,7,11,12,13,15], "Environment": "prod"}
```

```
1 CurrentSettings {"ActivityTypes": [0,1,2,3,4,5,6,7,13,15], "Environment": "prod"}
```

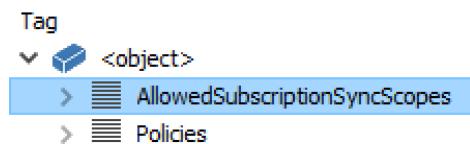
New Keys are introduced:

- DatabaseNotificationSubscriptionInfo,
- DatabaseActivityPolicies and
- PendingActivityTypes

as well as new ActivityTypes & Policies:

Key	Value
1 CurrentSettings	{"ActivityTypes": [0,1,2,3,4,5,6,7,11,12,13,15], "Environment": "prod"}
2 DatabaseInstanceId	23919
3 DatabaseInstanceIdUpdateTime	2018-05-10T13:58:59.758Z
4 CurrentEtag	ec9c6840-add6-11e8-9117-01020305070d
5 DatabaseNotificationSubscriptionInfo	{"partialSyncToken": "", "publisherFilters": [{"activityTypes": [0,1,2,3,4,5,6,7,13,15], "application": "*"}, {"activityTypes": [0,1,2,3,4,5,6,7,13,15], "application": "*"}]}
6 DatabaseActivityPolicies	{"AllowedSubscriptionSyncScopes": [{"allowedTypes": [0,1,2,3,4,5,6,7,10,13,15], "application": "*"}, {"allowedTypes": [0,1,2,3,4,5,6,7,10,13,15], "application": "*"}]}, {"Policies": [{"allowedTypes": [0,1,2,3,4,5,6,7,10,13,15], "application": "*"}, {"allowedTypes": [0,1,2,3,4,5,6,7,10,13,15], "application": "*"}]}]
7 PendingActivityTypes	{"activityTypes": []}

In the ‘DatabaseActivityPolicies’ key’s value (json field), we can see “AllowedSubscriptionSyncScopes” and “Policies” as the main fields:



“AllowedSubscriptionSyncScopes”’s Elements is where the Allowed Activity Types (*to be synched across devices*) are listed as well as the respective application & platform. For example:

Tag	Value
<object>	
AllowedSubscriptionSyncScopes	
<element #0>	
allowedTypes	
0	
1	
2	
3	
4	
5	
6	
7	
10	
13	
15	
application	*

ActivityTypes 0-7 and 10, 13 and 15 are allowed for (*) all applications, but

Tag	Value
<object>	
AllowedSubscriptionSyncScopes	
<element #0>	
<element #1>	
allowedTypes	
11	
12	
application	Microsoft.Credentials.WiFi
platform	data_boundary

Activity Types 11 and 12 are only allowed for the application “Microsoft.Credentials.WiFi” and only for the Platform entry “data_boundary”.

The Policies field

Tag	Value
Policies	
<element #0>	
<element #1>	
<element #2>	
<element #3>	
<element #4>	
<element #5>	
<element #6>	
<element #7>	

Has entries like this in each Element:

Tag	Value
<element #1>	
BlockedOperationFlags	0
Scope	
PermissionScope	Microsoft.Credentials
Type	11
Source	1
<element #2>	
BlockedOperationFlags	0
Scope	
PermissionScope	Microsoft.Personalization
Type	11
Source	1
<element #3>	
<element #4>	
BlockedOperationFlags	0
Scope	
PermissionScope	Microsoft.Credentials
Type	12
Source	1
<element #5>	
BlockedOperationFlags	0
Scope	
PermissionScope	Microsoft.Personalization
Type	12
Source	1

Where the PermissionScope is the Application, and Type is the ActivityType blocked for this application. BlockedOperationFlags has seen with values 0, 1 and 7. In the above screenshot, MicrosoftPersonalization and Microsoft.Credentials apps are NOT blocked for ActivityTypes 11 and 12.

BlockedOperationFlags value 1 means that this Element is Blocked for example, ActivityType 8 is blocked (*we can see that in the AllowedTypes as well*):

<element #25>	
BlockedOperationFlags	1
Scope	
PermissionScope	
Type	8
Source	7

Whereas value 0 means that this ActivityType is NOT blocked:

<element #11>	
└ BlockedOperationFlags	0
└ Scope	
└ PermissionScope	
└ Type	10
└ Source	4

(ActivityType 10 is listed in the AllowedTypes).

However a BlockedOperationFlags with value 7

<element #14>	
└ BlockedOperationFlags	7
└ Scope	
└ PermissionScope	Microsoft.Accessibility
└ Type	11
└ Source	5

Is not yet explained.

The key "DatabaseNotificationSubscriptionInfo" is also a JSON blob:

Tag	Value
<object>	
└ partialSyncToken	
└ publisherFilters	
└ subscriptionEtag	"11299795764551████████"
└ viewId	9e2d169e-6688-f677-dd22-4b████████

where in the 'publisherFilters' we see

 	publisherFilters	
 	<element #0>	
 	└ activityTypes	
 	└ 0	
 	└ 1	
 	└ 2	
 	└ 3	
 	└ 4	
 	└ 5	
 	└ 6	
 	└ 7	
 	└ 13	
 	└ 15	
 	└ application	*

that ActivityTypes 0-7, 13 and 15 are allowed for (*) all applications.

Finally 'PendingActivityTypes' value is also a JSON blob, but empty at this time.

Back to the ActivitiesCaches.db changes, there is also a new table, DataEncryptionKeys:

The screenshot shows the SQLite Database Browser interface. At the top, there are tabs: 'Database Structure' (disabled), 'Browse Data' (selected), 'Edit Pragmas', and 'Execute SQL'. Below the tabs, it says 'Table: DataEncryptionKeys'. The table has three columns: 'KeyVersion', 'KeyValue', and 'CreatedInCloudTime'. Each column has a 'Filter' button below it. There is one row of data: KeyVersion is 1, KeyValue is 'AQAAANCMnd8BFdERjHoAwE/Cl...', and CreatedInCloudTime is 1535762460.

KeyVersion	KeyValue	CreatedInCloudTime
1	3362556070382464122	AQAAANCMnd8BFdERjHoAwE/Cl... 1535762460

The Smartlookup view (query) view changed:

Smartlookup (1803)	Smartlookup (1809)
select	select
[O].[Id],	[O].[Id],
[O].[AppId],	[O].[AppId],
[O].[PackageIdHash],	[O].[PackageIdHash],
[O].[AppActivityId],	[O].[AppActivityId],
[O].[ActivityType],	[O].[ActivityType],
[O].[OperationType] AS [ActivityStatus],	[O].[OperationType] AS [ActivityStatus],
[O].[ParentActivityId],	[O].[ParentActivityId],
[O].[Tag],	[O].[Tag],
[O].[Group],	[O].[Group],
[O].[MatchId],	[O].[MatchId],
[O].[LastModifiedTime],	[O].[LastModifiedTime],
[O].[ExpirationTime],	[O].[ExpirationTime],
[O].[Payload],	[O].[Payload],
[O].[Priority],	[O].[Priority],
[A].[IsLocalOnly],	[A].[IsLocalOnly],
[O].[PlatformDeviceId],	[O].[PlatformDeviceId],
[A].[CreatedInCloud],	[A].[CreatedInCloud],
[O].[StartTime],	[O].[StartTime],
[O].[EndTime],	[O].[EndTime],
[O].[LastModifiedOnClient],	[O].[LastModifiedOnClient],
1 AS [IsInUploadQueue],	1 AS [IsInUploadQueue],
[O].[GroupAppActivityId],	[O].[GroupAppActivityId],
[O].[ClipboardPayload],	[O].[ClipboardPayload],
[O].[EnterpriseId],	[O].[EnterpriseId],
[O].[OriginalPayload],	[O].[UserActionState],
[O].[OriginalLastModifiedOnClient],	[O].[IsRead],
[O].[ETag]	[O].[OriginalPayload],
from [ActivityOperation] as [O]	[O].[OriginalLastModifiedOnClient],
left outer join [Activity] as [A] on [O].[Id] = [A].[Id]	[O].[GroupItems],
union	[O].[ETag]
select	from [ActivityOperation] as [O]
[Id],	left outer join [Activity] as [A] on [O].[Id] = [A].[Id]
[AppId],	union
[PackageIdHash],	select
[AppActivityId],	[Id],
[ActivityType],	[AppId],
[ActivityStatus],	[PackageIdHash],
[ParentActivityId],	[AppActivityId],
[Tag],	[ActivityType],
[Group],	[ActivityStatus],
[MatchId],	[ParentActivityId],
[LastModifiedTime],	[Tag],
[ExpirationTime],	[Group],
[Payload],	[MatchId],
[Priority],	[LastModifiedTime],
[IsLocalOnly],	[ExpirationTime],
[PlatformDeviceId],	[Payload],
[CreatedInCloud],	[Priority],
[StartTime],	[IsLocalOnly],
[EndTime],	[PlatformDeviceId],

[LastModifiedOnClient],	[CreatedInCloud],
0 AS [IsInUploadQueue],	[StartTime],
[GroupAppActivityId],	[EndTime],
[ClipboardPayload],	[LastModifiedOnClient],
[EnterpriseId],	0 AS [IsInUploadQueue],
[OriginalPayload],	[GroupAppActivityId],
[OriginalLastModifiedOnClient],	[ClipboardPayload],
[ETag]	[EnterpriseId],
from [Activity]	[UserActionState],
where [Id] not in (select [Id]	[IsRead],
from [ActivityOperation])	[OriginalPayload],
	[OriginalLastModifiedOnClient],
	[GroupItems],
	[ETag]
	from [Activity]
	where [Id] not in (select [Id]
	from [ActivityOperation])

We also see a new PlatformID entry: "data_boundary"

ETag	p0	p1	p2	p3
127201	data_boundary	packageId	alternateId	NULL
127203	data_boundary	packageId	alternateId	NULL
127205	data_boundary	packageId	alternateId	NULL
127207	data_boundary	packageId	alternateId	NULL
127209	data_boundary	packageId	alternateId	NULL
127211	data_boundary	packageId	alternateId	NULL
127213	data_boundary	packageId	alternateId	NULL

Associated with Activity types 11 and 12:

AppId	PackageldHash	AppActivityId	Activity type	ActivityStatus
{"application": "Microsoft.Credentials.WiFi", "platform": "data_boundary"},...	YNLZe4hPeNt2TsD3m2/J...	default\$windows.data.nl...	11	Active
{"application": "Microsoft.Credentials.WiFi", "platform": "data_boundary"},...	YNLZe4hPeNt2TsD3m2/J...	default\$windows.data.nl...	11	Active
{"application": "Microsoft.Credentials.WiFi", "platform": "data_boundary"},...	YNLZe4hPeNt2TsD3m2/J...	default\$windows.data.nl...	11	Active
{"application": "Microsoft.Credentials.WiFi", "platform": "data_boundary"},...	YNLZe4hPeNt2TsD3m2/J...	default\$windows.data.nl...	11	Active
{"application": "Microsoft.Credentials.WiFi", "platform": "data_boundary"},...	YNLZe4hPeNt2TsD3m2/J...	default\$windows.data.nl...	11	Active
{"application": "Microsoft.Credentials.WiFi", "platform": "data_boundary"},...	YNLZe4hPeNt2TsD3m2/J...	default\$windows.data.wifi...	12	Updated
{"application": "Microsoft.Credentials.WiFi", "platform": "data_boundary"},...	YNLZe4hPeNt2TsD3m2/J...	default\$windows.data.wifi...	12	Updated

Whose payload is encoded/encrypted.

And one entry associated with Activity type 15:

AppId	PackageldHash	AppActivityId	Activity type
[{"application": "dek.encrypted.settings", "platform": "alternateId"}, {"application": "", "..."}]	aFEhkMTLDtGKXWcGe0U...		15

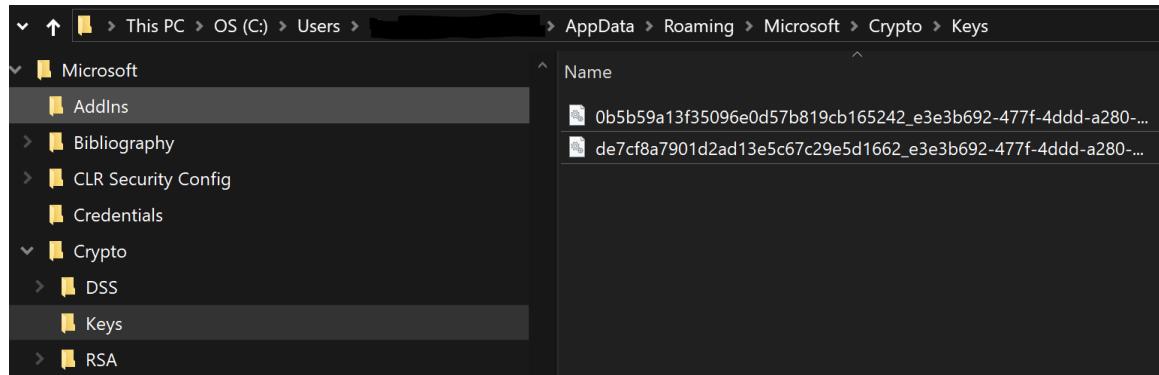
whose Payload field is encoded with Base64, and encrypted. Using [`https://gchq.github.io/CyberChef/#recipe=From_Base64\('A-Za-z0-9%2B/%3D',true\)`](https://gchq.github.io/CyberChef/#recipe=From_Base64('A-Za-z0-9%2B/%3D',true)) I was able to see the email (*Microsoft account*) associated with this Win 10 installation.

In the ActivityOperation table, a new column (field) named “UploadAllowedByPolicy” is introduced:

ETag	UploadAllowedByPolicy
Filter	Filter
130207	1
130208	1

With observed values 0 and 1. The obvious deduction is that entries with value 1 are Yes and entries with value 0 are No. The Metadata table is where these policies are defined.

Note: The encryption (?) certificates are stored at
C:\Users\%Username%\AppData\Roaming\Microsoft\Crypto\Keys



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
000000000	D1	00	00	00	00	00	00	00	6E	00	00	00	04	00	03	00n.....
000000010	88	00	00	00	42	01	00	00	5C	01	00	00	00	00	00	00B...\\.....
000000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00M.i.....
000000030	63	00	72	00	6F	00	73	00	6F	00	66	00	74	00	20	00	c.r.o.s.o.f.t. .
000000040	43	00	6F	00	6E	00	6E	00	65	00	63	00	74	00	65	00	C.o.n.n.e.c.t.e. .
000000050	64	00	20	00	44	00	65	00	76	00	69	00	63	00	65	00	d. .D.e.v.i.c.e. .
000000060	73	00	20	00	50	00	6C	00	61	00	74	00	66	00	6F	00	s. .P.l.a.t.f.o. .
000000070	72	00	6D	00	20	00	64	00	65	00	76	00	69	00	63	00	r.m. .d.e.v.i.c. .
000000080	65	00	20	00	63	00	65	00	72	00	74	00	69	00	66	00	e. .c.e.r.t.i.f. .
000000090	69	00	63	00	61	00	74	00	65	00	2C	00	00	00	00	00	i.c.a.t.e.,.....
0000000A0	00	00	00	00	00	00	10	00	00	00	08	00	00	00	4D	00M. .
0000000B0	6F	00	64	00	69	00	66	00	69	00	65	00	64	00	FD	01	o.d.i.f.i.e.d.ú. .
0000000C0	B5	F7	94	4D	D4	01	5C	00	00	00	0A	00	00	00	00	00	µX"MT.\\.....
0000000D0	00	00	00	00	00	00	48	00	00	00	45	43	53	31	20	00H..ECS1 .
0000000E0	00	00	7E	86	DA	2B	38	59	59	CE	80	D2	F2	52	EF	F8	..~†Í+YYΞ€□çRoΨ .
0000000F0	0C	43	D9	33	3A	13	9A	20	6F	8D	B5	30	F5	B6	9C	FE	:CΩ3:.. o.µ0ü .ω .
00000100	27	D4	2F	78	30	F8	F1	47	6B	DF	94	21	74	B0	08	5A	'T/x0ΨρGki"!t°.Z .
00000110	E9	89	C5	EE	3E	99	4E	74	8D	1D	BA	0C	EA	01	3E	E8	L%EΞ>™Nt..T.k.>θ .
00000120	69	62	01	00	00	00	D0	8C	9D	DF	01	15	D1	11	8C	7A	ib....Π..ι..P..z .
00000130	00	C0	4F	C2	97	EB	01	00	00	D5	F7	ED	6E	03	BE	.ÍOB-λ....Yχvn.Y .	
00000140	02	4E	8F	4D	ED	4B	18	4A	8A	8A	00	00	00	00	2E	00	.N.MvK.j.....
00000150	00	00	50	00	72	00	69	00	76	00	61	00	74	00	65	00	..P.r.i.v.a.t.e. .
00000160	20	00	4B	00	65	00	79	00	20	00	50	00	72	00	6F	00	.K.e.y. .P.r.o. .
00000170	70	00	65	00	72	00	74	00	69	00	65	00	73	00	00	00	p.e.r.t.i.e.s... .
00000180	10	66	00	00	00	01	00	00	20	00	00	00	1A	48	D4	58	.f..... .HTX .
00000190	FA	0E	89	9F	16	55	E1	82	19	CD	8D	24	85	EF	BC	D9	L.%..Uø,.N.\$..o0Ω .
000001A0	B2	E5	1E	7F	C3	7E	25	1C	E8	FA	03	FE	00	00	00	00	2ε..Γ~%.θL.ω.... .

Update (Win 10 version 1903 OS Build 18875.1000)

In the latest update, we have a new “ActivityType” -> 10 which is clipboard data (I have only observed clipboard text).

Clipboard

Clipboard history

Save multiple items to the clipboard to use later. Press the Windows logo key + V to view your clipboard history and paste from it.



On

Sync across devices

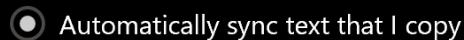
Paste text on your other devices. When this is on, Microsoft receives your clipboard data to sync it across your devices.



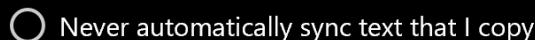
On

Get an app to sync clipboard items to your phone

Automatic syncing



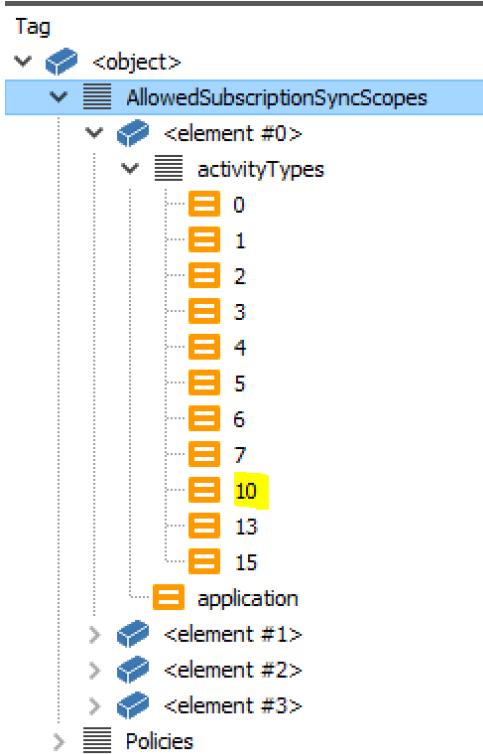
Text copied to the clipboard is synced to your other devices.



Open clipboard history (Windows logo key + V) to choose text to sync.

The Metadata table now includes ActivityType 10 :

Table: Metadata	
	Key
Filter	Filter
1 DatabaseNotificationSubscriptionInfo	{"ddsDeviceId":"","partialSyncToken":"","publisherFilters":[{"activityTyp
2 PendingActivityTypes	{"activityTypes":[]}
3 DatabaseActivityPolicies	{"AllowedSubscriptionSyncScopes":[{"activityTypes":[0,1,2,3,4,5,6,7,10,13
4 CurrentSettings	{"ActivityTypes":[0,1,2,3,4,5,6,7,10,11,13,15],"Environment":"prod"}
5 PendingFirstDEKUpload	false
6 CurrentEtag	cbacde20-5d0e-11e9-9117-01020305070d
7 DatabaseInstanceId	23919
8 DatabaseInstanceIdUpdateTime	2018-05-10T13:58:59.758Z



The field “ClipboardPayload” (JSON blob) of entries with ActivityType 10 (Clipboard text) contain the copied text in Base64 encoding in the form:

```
[{"content": "eyJ..."}]
```

The JSON blob contains a single object with a 'content' key. The value of 'content' is a long string of Base64 encoded text. The string starts with 'eyJBbGxvd2VkJ...'. It is a very long and complex string, likely representing a large amount of clipboard text copied from a document or application.

sIIr5cGUIojeYfSwiU291cmNIIjozfSx7Ikjsb2NrZWRPcGVyYXRpb25GbGFncyl6NywiU2NvcGUIOnsiUGVybWlzaW9uU2NvcGUIoiiLCJUeXBIIjoxNX0sINvdXjzSI6M30seyJCbG9ja2Vkt3BlcmF0aW9uRmxhZ3MiOjcsIINjb3BlIjp7IIBlcm1pc2lvbINjb3BlIjoiwiVHlwZSI6MTB9LCJTb3VY2UiOjR9LHSiQmxvY2tlZE9wZXJhdGlvbkzsYwdzljowLCJTY29wZSI6eyJQZXJtaXNpb25TY29wZSI6lilsIIr5cGUIojeFwfSwiU291cmNIIjo1fSx7Ikjsb2NrZWRPcGVyYXRpb25GbGFncyl6MCwiU2NvcGUIOnsiUGVybWlzaW9uU2NvcGUIoiiLCJUeXBIIjoxMX0sINvdXjzSI6NX0seyJCbG9ja2Vkt3BlcmF0aW9uRmxhZ3MiOjcsIINjb3BlIjp7IIBlcm1pc2lvbINjb3BlIjoiTWIjc9zb2Z0LkFjY2Vzc2liaWxpdHkiLCJUeXBIIjoxMX0sINvdXjzSI6NX0seyJCbG9ja2Vkt3BlcmF0aW9uRmxhZ3MiOjcsIINjb3BlIjp7IIBlcm1pc2lvbINjb3BlIjoiTWIjc9zb2Z0LkNyZWRlbnRpYWxzliwiVHlwZSI6MTF9LCJTb3VY2UiOjV9LHSiQmxvY2tlZE9wZXJhdGlvbkzsYwdzljowLCJTY29wZSI6eyJQZXJtaXNpb25TY29wZSI6Ik1pY3Jvc29mdC5EZWZhdWx0liwiVHlwZSI6MTF9LCJTb3VY2UiOjV9LHSiQmxvY2tlZE9wZXJhdGlvbkzsYwdzljowLCJTY29wZSI6eyJQZXJtaXNpb25TY29wZSI6Ik1pY3Jvc29mdC5MYW5ndWFnZIsIIr5cGUIojeFwfSwiU291cmNIIjo1fSx7Ikjsb2NrZWRPcGVyYXRpb25GbGFncyl6NywiU2NvcGUIOnsiUGVybWlzaW9uU2NvcGUIoijNaWNyb3NvZnQuUGVyc29uYWxpemF0aW9uliwiVHlwZSI6MTF9LCJTb3VY2UiOjV9LHSiQmxvY2tlZE9wZXJhdGlvbkzsYwdzljowLCJTY29wZSI6eyJQZXJtaXNpb25TY29wZSI6lilsIIr5cGUIojeFwfSwiU291cmNIIjo1fSx7Ikjsb2NrZWRPcGVyYXRpb25GbGFncyl6NywiU2NvcGUIOnsiUGVybWlzaW9uU2NvcGUIoijNaWNy b3NvZnQuQWNjZXNzaWjbGl0eSIsIIr5cGUIojeFwfSwiU291cmNIIjo1fSx7Ikjsb2NrZWRPcGVyYXRpb25GbGFncyl6NywiU2NvcGUIOnsiUGVybWlzaW9uU2NvcGUIoijNaWNyb3NvZnQuQ3JIZGVudGihbHMiLCJUeXBIIjoxMn0sINvdXjzSI6NX0seyJCbG9ja2Vkt3BlcmF0aW9uRmxhZ3MiOjAsIINjb3BlIjp7IIBlcm1pc2lvbINjb3BlIjoiTWIjcm9zb2Z0LkRIzmf1bHQiLCJUeXBIIjoxMn0sINvdXjzSI6NX0seyJCbG9ja2Vkt3BlcmF0aW9uRmxhZ3MiOjcsIINjb3BlIjp7IIBlcm1pc2lvbINjb3BlIjoiTWIjcm9zb2Z0Lkxhbmd1YWdliwiVHlwZSI6MTJ9LCJTb3VY2UiOjV9LHSiQmxvY2tlZE9wZXJhdGlvbkzsYwdzljowLCJTY29wZSI6eyJQZXJtaXNpb25TY29wZSI6Ik1pY3Jvc29mdC5QZXJbz25hbGl6YXRpb24iLCJUeXBIIjoxMn0sINvdXjzSI6NX0seyJCbG9ja2Vkt3BlcmF0aW9uRmxhZ3MiOjEsIINjb3BlIjp7IIBlcm1pc2lvbINjb3BlIjoiwiVHlwZSI6OH0sINvdXjzSI6N30seyJCbG9ja2Vkt3BlcmF0aW9uRmxhZ3MiOjEsIINjb3BlIjp7IIBlcm1pc2lvbINjb3BlIjoiwiVHlwZSI6OX0sINvdXjzSI6N30seyJCbG9ja2Vkt3BlcmF0aW9uRmxhZ3MiOjAsIINjb3BlIjp7IIBlcm1pc2lvbINjb3BlIjoiwiVHlwZSI6MTB9LCJTb3VY2UiOjd9LHSiQmxvY2tlZE9wZXJhdGlvbkzsYwdzljowLCJTY29wZSI6eyJQZXJtaXNpb25TY29wZSI6lilsIIr5cGUIojeFwfSwiU291cmNIIjo3fSx7Ikjsb2NrZWRPcGVyYXRpb25GbGFncyl6MCwiU2NvcGUIOnsiUGVybWlzaW9uU2NvcGUioiliLCJUeXBIIjoxMn0sINvdXjzSI6N30seyJCbG9ja2Vkt3BlcmF0aW9uRmxhZ3MiOjAsIINjb3BlIjp7IIBlcm1pc2lvbINjb3BlIjoiwiVHlwZSI6MTB9LCJTb3VY2UiOjd9LHSiQmxvY2tlZE9wZXJhdGlvbkzsYwdzljowLCJTY29wZSI6eyJQZXJtaXNpb25TY29wZSI6lilsIIr5cGUIojeFwfSwiU291cmNIIjo1fSx7Ikjsb2NrZWRPcGVyYXRpb25GbGFncyl6MCwiU2NvcGUIOnsiUGVybWlzaW9uU2NvcGUioiliLCJUeXBIIjoxMn0sINvdXjzSI6N30seyJCbG9ja2Vkt3BlcmF0aW9uRmxhZ3MiOjAsIINjb3BlIjp7IIBlcm1pc2lvbINjb3BlIjoiwiVHlwZSI6MTB9LCJTb3VY2UiOjd9LHSiQmxvY2tlZE9wZXJhdGlvbkzsYwdzljowLCJTY29wZSI6eyJQZXJtaXNpb25TY29wZSI6lilsIIr5cGUIojeFwfSwiU291cmNIIjo5fV19", "formatName": "Text"}]

content	T W3siY29udGVudCI6ImV5SkJiR3h2ZDJWa1UzVm1jMk55YVh0
formatName	T Text

(Translated below:)

Recipe

From Base64

Alphabet
A-Za-z0-9+=

Remove non-alphabet chars

Input

```
eyJ0BbGxvd2Vku3Vic2NyaXB0aw9uU3luY1Njb3BlcyI6W3siYWN0aXZpdHlUeXBlcI6WzAsMSwyLDMsNCw1LDysNywxMcwxMyw
xNV0sImFwcGxpY2F0aw9uIjoikIj9LhsiyWN0aXZpdHlUeXBlcI6WzExLEyXSwiYXBwbGljYXRpb24iOjNaWNyb3NvZnQuQ3
J1ZGVudGhbHMuvmF1bHQiLCJwbGF0Zm9ybSI6ImRhdGFFYm91bmRhcnkifSx7ImFjdG12aXR5VHlwZXMiolsxM5wxM10sImFwc
GxpY2F0aw9uIjoitWljcm9zb2Z0LkNyZWRlbnRpYnxzLldpRmkilCJwbGF0Zm9ybSI6ImRhdGFFYm91bmRhcnkifSx7ImFjdG12
aXR5VHlwZXMiolsxMV0sImFwcGxpY2F0aw9uIjoibWljcm9zb2Z0LmR1ZmF1bHQzGvMvXVsdcIsInBsYXRmb3JtIjoizGF0YV9
ib3VuZGFyeSJ9XSwiUG9saWNpZXMiolt7Ikjsb2NrZWRPcGVyYXRpb25GbGFncyI6MCwiU2NvcGUiOnsiUGVybW1zaW9uU2NvcG
Ui0iiilCJuexBlijozMx0sIlNvdXJjZSI6MX0seyJCbG9ja2Vkt3Blcmf0aw9uRmxhZ3MiojAsIljb3BlIip7IlBlcm1pc2lvb
```

Output

```
{"AllowedSubscriptionSyncScopes": [{"activityTypes": [0, 1, 2, 3, 4, 5, 6, 7, 10, 13, 15], "application": "*"}, {"activityTypes": [11, 12], "application": "Microsoft.Credentials.Vault", "platform": "data_boundary"}, {"activityTypes": [11, 12], "application": "Microsoft.Credentials.WiFi", "platform": "data_boundary"}, {"activityTypes": [11], "application": "microsoft.default.default", "platform": "data_boundary"}], "Policies": [{"BlockedOperationFlags": 0, "Scope": {"PermissionScope": "", "Type": 11}, "Source": 1}, {"BlockedOperationFlags": 0, "Scope": {"PermissionScope": "Microsoft.Credentials", "Type": 11}, "Source": 1}, {"BlockedOperationFlags": 0, "Scope": {"PermissionScope": "Microsoft.Personalization", "Type": 11}, "Source": 1}, {"BlockedOperationFlags": 0, "Scope": {"PermissionScope": "", "Type": 12}, "Source": 1}, {"BlockedOperationFlags": 0, "Scope": {"PermissionScope": "Microsoft.Credentials", "Type": 12}, "Source": 1}, {"BlockedOperationFlags": 0, "Scope": {"PermissionScope": "Microsoft.Personalization", "Type": 12}, "Source": 1}, {"BlockedOperationFlags": 0, "Scope": {"PermissionScope": "", "Type": 11}, "Source": 2}, {"BlockedOperationFlags": 0, "Scope": {"PermissionScope": "", "Type": 12}, "Source": 2}, {"BlockedOperationFlags": 7, "Scope": {"PermissionScope": "", "Type": 10}, "Source": 3}], "start: 0 end: 2 length: 4712 lines: 1", "start: 0 time: 5ms end: 1 length: 3534 lines: 1", "start: 0 end: 1 length: 1 lines: 1"}
```

The payload of the above entry is:

```
{"gdprType": "ProductAndServiceUsage", "clipboardDataId": "B268A259-DAAD-47FA-A089-55E3BB298BCB"}
```

Another new 'ActivityType' 16 contains data on the application from where data was copied/pasted from/to, and the 'Group' field shows what action was operated:

AppActivityId	ActivityType	ActivityStatus	arentActivityId	Tag	Group
Filter	Filter	Filter	Filter	Filter	Filter
ECB32AF3-1440...	16	1	BLOB	NULL	Copy
ECB32AF3-1440...	6	2	BLOB	NULL	NULL

And where it was pasted:

AppId	PackageIdHash	AppActivityId	ActivityType	ActivityStatus	arentActivityId	Tag	Group
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
[{"applicatio... Rg8DLH4/nR45L...			10	1	BLOB	NULL	NULL
[{"applicatio... deglsgq0fpSkY...	ECB32AF3-1440...	6	1	BLOB	NULL	NULL	NULL
[{"applicatio... bwfyZ5c003AQa...	ECB32AF3-1440...	16	1	BLOB	NULL	Paste	
[{"applicatio... Ts0+SH7DZfSHh...	ECB32AF3-1440...	6	1	BLOB	NULL	NULL	

We can now see copy pasted operations:

Group	GroupAppActivityId	GroupItems	Is_Read	EnterpriseId	ParentActivityId
Paste			No		4233DA52CE1FF43168B53...
Copy			No		EF261D0B550FC22F26BD2...
Copy			No		EF261D0B550FC22F26BD2...
Paste			No		4233DA52CE1FF43168B53...
Copy			No		EF261D0B550FC22F26BD2...
Copy			No		EF261D0B550FC22F26BD2...
Paste			No		4233DA52CE1FF43168B53...
Copy			No		EF261D0B550FC22F26BD2...
Copy			No		EF261D0B550FC22F26BD2...
Paste			No		4233DA52CE1FF43168B53...
Copy			No		EF261D0B550FC22F26BD2...
Copy			No		EF261D0B550FC22F26BD2...
Paste			No		4233DA52CE1FF43168B53...
Copy			No		EF261D0B550FC22F26BD2...
Copy			No		EF261D0B550FC22F26BD2...

The copy paste operations can be linked with the 'clipboardDataId' value in the Payload field (JSON) eg:

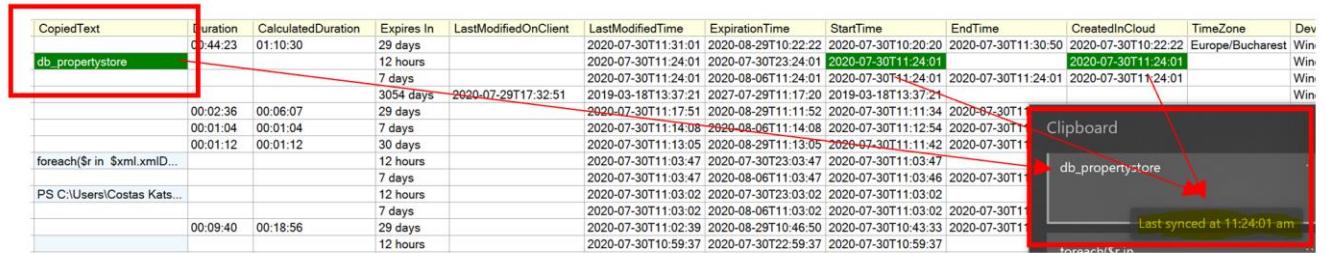
```
{"gdprType":"ProductAndServiceUsage","clipboardDataId":"{BA306A6E-2B7B-483B-B9B6-76BB7AF64D12}"}
```

Activity_type	Group	GroupAppActivityId	GroupItems	ParentActivityId	DdsDeviceId	Device ID	Expires In days	clipboardDataId
Copy/Paste	Paste			A7A0DBC58DD13A75E991D...	NULL	dpRc3VX485JVH95n3ZaaJ...	31	{BA306A6E-2B7B-483B-B9B6-76BB7AF64D1...
Clipboard	NULL			00000000000000000000...	NULL	dpRc3VX485JVH95n3ZaaJ...	0	NULL
Copy/Paste	Copy			9B1C816547035726A2181...	NULL	dpRc3VX485JVH95n3ZaaJ...	31	{BA306A6E-2B7B-483B-B9B6-76BB7AF64D1...
Clipboard	NULL			00000000000000000000...	NULL	dpRc3VX485JVH95n3ZaaJ...	0	NULL
Copy/Paste	Copy			A7A0DBC58DD13A75E991D...	NULL	dpRc3VX485JVH95n3ZaaJ...	31	{91FAD9DC-6A43-475C-8034-7929148F41E...
Clipboard	NULL			00000000000000000000...	NULL	dpRc3VX485JVH95n3ZaaJ...	0	NULL
Copy/Paste	Copy			78009A0829A4BAD23D5A8...	NULL	dpRc3VX485JVH95n3ZaaJ...	31	{48A8B44B-6EF6-49F7-8742-A0BAF8F8D35...
Copy/Paste	Paste			78009A0829A4BAD23D5A8...	NULL	dpRc3VX485JVH95n3ZaaJ...	31	{3DB5D8D1-0518-4B03-AD8D-082F6BF3451...
Copy/Paste	Paste			78009A0829A4BAD23D5A8...	NULL	dpRc3VX485JVH95n3ZaaJ...	31	{3DB5D8D1-0518-4B03-AD8D-082F6BF3451...
Copy/Paste	Paste			78009A0829A4BAD23D5A8...	NULL	dpRc3VX485JVH95n3ZaaJ...	31	{3DB5D8D1-0518-4B03-AD8D-082F6BF3451...
Copy/Paste	Paste			78009A0829A4BAD23D5A8...	NULL	dpRc3VX485JVH95n3ZaaJ...	31	{3DB5D8D1-0518-4B03-AD8D-082F6BF3451...
Copy/Paste	Paste			78009A0829A4BAD23D5A8...	NULL	dpRc3VX485JVH95n3ZaaJ...	31	{3DB5D8D1-0518-4B03-AD8D-082F6BF3451...
Copy/Paste	Paste			78009A0829A4BAD23D5A8...	NULL	dpRc3VX485JVH95n3ZaaJ...	31	{3DB5D8D1-0518-4B03-AD8D-082F6BF3451...
Clipboard	NULL			00000000000000000000...	NULL	dpRc3VX485JVH95n3ZaaJ...	0	NULL
Copy/Paste	Copy			78009A0829A4BAD23D5A8...	NULL	dpRc3VX485JVH95n3ZaaJ...	31	{3DB5D8D1-0518-4B03-AD8D-082F6BF3451...

The Clipboard text has a duration (ExpirationTime - LastModifiedTime) of 43200 seconds or 12 hours exactly.

Text(Base64)	ClipboardPayload	Activity_type	Group	GroupAppActivityId	GroupItems	ParentActivityId	DdsDeviceId	Device ID	Expires In days
U09GVFdBUkVcQ2xpZW50c...	["content":"U09GVFdB...	Copy/Paste	Clipboard	NULL	A7A0DBC58DD13A75E991D...	NULL	dpRc3VX485J... 31	dpRc3VX485J... 31	dpRc3VX485J... 0
MzA4MDQ2QjBBRjRBMz1DQ...	["content":"MzA4MDQ2...	Copy/Paste	Clipboard	NULL	9B1C816547035726A2181...	NULL	dpRc3VX485J... 31	dpRc3VX485J... 31	dpRc3VX485J... 0
PGJyPg==	["content":"PGJyPg==	Copy/Paste	Clipboard	NULL	A7A0DBC58DD13A75E991D...	NULL	dpRc3VX485J... 31	dpRc3VX485J... 31	dpRc3VX485J... 0
PGJyPg==	["content":"PGJyPg==	Copy/Paste	Clipboard	NULL	00000000000000000000...	NULL	dpRc3VX485J... 31	dpRc3VX485J... 31	dpRc3VX485J... 0
PGJyPg==	["content":"PGJyPg==	Copy/Paste	Clipboard	NULL	78009A0829A48AD23D5A8...	NULL	dpRc3VX485J... 31	dpRc3VX485J... 31	dpRc3VX485J... 31
PGJyPg==	["content":"PGJyPg==	Copy/Paste	Clipboard	NULL	78009A0829A48AD23D5A8...	NULL	dpRc3VX485J... 31	dpRc3VX485J... 31	dpRc3VX485J... 31
PGJyPg==	["content":"PGJyPg==	Copy/Paste	Clipboard	NULL	78009A0829A48AD23D5A8...	NULL	dpRc3VX485J... 31	dpRc3VX485J... 31	dpRc3VX485J... 0
DQogICAgV2luZG93cyJB...	["content":"DQogICAg...	Copy/Paste	Clipboard	NULL	78009A0829A48AD23D5A8...	NULL	dpRc3VX485J... 31	dpRc3VX485J... 31	dpRc3VX485J... 31
Q1NleDJTWTFoCHBkNWxOT...	["content":"Q1NleDJT...	Copy/Paste	Clipboard	NULL	78009A0829A48AD23D5A8...	NULL	dpRc3VX485J... 31	dpRc3VX485J... 31	dpRc3VX485J... 0
Q1NleDJTWTFoCHBkNWxOT...	["content":"Q1NleDJT...	Copy/Paste	Clipboard	NULL	00000000000000000000...	NULL	dpRc3VX485J... 31	dpRc3VX485J... 31	dpRc3VX485J... 0

The Copied text can also be seen by the user by pressing the Windows key + V. Pointing the mouse arrow to one of the copied text items, one can see the Last Synced time of that item in a tooltip:



Note: Images, as they aren't syncing (yet), do not behave this way. I.e. if you point the mouse arrow to a copied image, there is no 'Last synced' timestamp tooltip.

Another location with full CDP info

C:\Windows\ServiceProfiles\LocalService\AppData\Local\ConnectedDevicesPlatform

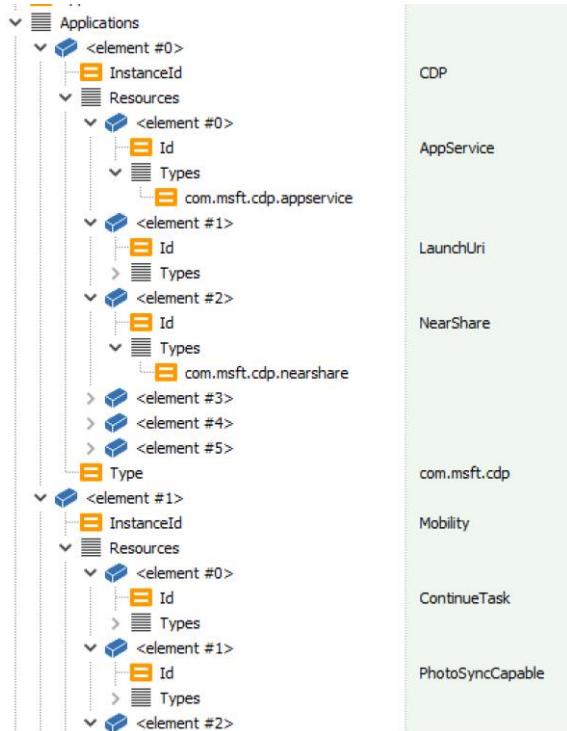
Name	Date modified	Type	Size
0ffc25d8321343c3.cdp	29 Feb 20 9:58 pm	CDP File	30 KB
3d9ee62b8d4d0071.cdp	29 Feb 20 9:58 pm	CDP File	33 KB
4a7cc0f3b728d240.cdp	29 Feb 20 9:58 pm	CDP File	44 KB
AAD.e5316abf-be67-4d21-9662-d64159c3608a...	29 Feb 20 9:58 pm	CDP File	13 KB
CDPGlobalSettings.cdp	29 Feb 20 9:58 pm	CDP File	53 KB
Connected Devices Platform certificates.sst	30 May 18 12:41 pm	Microsoft Serialized ...	25 KB
LCostas Katsavounidis.cdp	29 Feb 20 9:58 pm	CDP File	1 KB

Tag	Value
<object>	
> AfcDatabaseSettings	
AfsActivityTypes	
AfsChannelUri	
AfsEnvironment	
AfsSubscriptionId	
AfsSubscriptionUpdateTime	0000-00-00T00:00:00.000
BaseRegisteredInfoHash	E73hmQh3crvrucgsIjeCzsN192P+k9P+r
CNCNotificationUri	
CNCNotificationUriExpirationTime	0000-00-00T00:00:00.000
CNCNotificationUriLastSynced	0000-00-00T00:00:00.000
DdsRegistrationExpiryTickCount	15974007717148974
< Devices >	
> <element #0>	
> <element #1>	
> <element #2>	
> <element #3>	
> <element #4>	
> <element #5>	
> <element #6>	
FormatVersion	12
LastRegisteredNotificationUri	https://am3p.notify.windows.com/?token
LastRegisteredNotificationUriExpirationTime	2020-08-15T10:26:09.000Z
LastSyncedTime	2020-07-21T17:35:29.071Z
LogicalDeviceId	
NeedsToRevokeChannel	false
NextDataEncryptionKeyRolloverTime	0000-00-00T00:00:00.000
RegisteredInfoHash	2eH752D1Xsdaa4+akNzMLSGksVHLBsFC
RegisteredWithStrongAuth	false
StableUserId	4e-3e-3f-3d-40

Full Device Info:

Children: 22 Level: 2	
Name	Value
ApplicationDisplayName	
ApplicationId	
Applications	(array)
BluetoothClassicMacAddress	
CcsTicket	.eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dC16ImVpSlpCcJodHRwOi8vZGRzLiwiaXNzIjoiQ04nd0c2lrbmluZy0ylmRkcywlCJleHAQJE10TC5NDU1MzAsInNwc2kiOURd20wbndXN0NSI2020-08-20T17:45:30.916Z
CcsTicketExpiration	
DeduplicationHint	
DeviceDdsClientId	530a8b79-f55a-5d7a-897a-36603253771d
DeviceThumbprint	g3dnHVFMIVc6l3zt531jiysStzfpDRxNAuCc//Ceer0=
IPv4Address	192.168.1.143
ManufacturerName	DellInc.
ModelName	Venue 8 Pro 5830
ShortOSVersion	10.0.19041.330
UserDeviceThumbprint	DFw3s8u7srTt1N++HfHJkrdN+RPuR+D6EnPdhJRhv40=
connectionModes	4
deviceId	g3dnHVFMIVc6l3zt531jiysStzfpDRxNAuCc//Ceer0=
endpoints	(array)
extendedDeviceStatus	0
isMyDevice	true
name	Dell_Venue8
status	3
type	16

And capability info:



A quick PackageIdHash test:

I made a copy of an executable on my desktop & renamed it. Here are the hashes of the two apps, showing they are identical:

Filename	MD5	SHA1	CRC32
D:\ODC FSD info test app.exe	9DBFF96D5CF7FDC4CD0C5285D3CB1844	861ED00E54296AAE27AC66F8EAAED103249...	733EF785
D:\ODCFSDInfo.exe	9DBFF96D5CF7FDC4CD0C5285D3CB1844	861ED00E54296AAE27AC66F8EAAED103249...	733EF785

And here is the relevant info from ActivitiesCache.db

ETag	Application	Display Name	File Opened
809375	D:\Costas\Desktop\ODCFSDInfo.exe		
809374	D:\Costas\Desktop\ODCFSDInfo.exe	ODCFSDInfo.exe	ODCFSDInfo.exe
809373	D:\Costas\Desktop\ODC FSD info test app.exe		
809369	D:\Costas\Desktop\ODC FSD info test app.exe		
809368	D:\Costas\Desktop\ODC FSD info test app.exe	ODC FSD info test app.exe	ODC FSD info test app.exe

Showing a mismatch in the PackageIdHash value:

PackageldHash
O0SlgddmGo2acCj6LWmJ61RsdMpKy5UH8/AeED0hoCA=
O0SlgddmGo2acCj6LWmJ61RsdMpKy5UH8/AeED0hoCA=
5LA3QnQhXxhsvUapRxFKbLI0rNjwrEM+PbtfJfdDTv8=
5LA3QnQhXxhsvUapRxFKbLI0rNjwrEM+PbtfJfdDTv8=
5LA3QnQhXxhsvUapRxFKbLI0rNjwrEM+PbtfJfdDTv8=

Note, the original app's PackageIdHash is
“O0SlgddmGo2acCj6LWmJ61RsdMpKy5UH8/AeED0hoCA=”

I deleted the original ODCFSDInfo.exe and renamed the second app as “ODCFSDInfo.exe”. I run that app a few times to make sure I won't confuse the entries with the previous ones

ETag	Application	PackageldHash
809587	D:\Costas\Desktop\ODCFSDInfo.exe	O0SlgddmGo2acCj6LWmJ61RsdMpKy5UH8/AeED0hoCA=
809577	D:\Costas\Desktop\ODCFSDInfo.exe	O0SlgddmGo2acCj6LWmJ61RsdMpKy5UH8/AeED0hoCA=
809575	D:\Costas\Desktop\ODCFSDInfo.exe	O0SlgddmGo2acCj6LWmJ61RsdMpKy5UH8/AeED0hoCA=
809571	D:\Costas\Desktop\ODCFSDInfo.exe	O0SlgddmGo2acCj6LWmJ61RsdMpKy5UH8/AeED0hoCA=

Now the PackageldHash is the same as the original file's.

Copied the original app to another folder:

Full Path	MD5	SHA1
D:\Costas\Desktop\ODCFSDInfo.exe	9DBFF96D5CF7FDC4CD0C5285D3CB1844	861ED00E54296AAE27AC66F8EAAED103249...
D:\Temp\ODCFSDInfo.exe	9DBFF96D5CF7FDC4CD0C5285D3CB1844	861ED00E54296AAE27AC66F8EAAED103249...

executed it

ETag	Application	Display Name	File Opened
809828	D:\Temp\ODCFSDInfo.exe	ODCFSDInfo.exe	ODCFSDInfo.exe

And got a totally different PackageIdHash this time:

PackageIdHash
KaSajPw+fmzFuXVPSLWHzyMTkMH6P93d1RhcG+NNDXg=

So, the path is as important as is the filename for the hash.

Edited 2 bytes of the file

Filename	Full Path	MD5	SHA1
ODCFSDinfo.exe	D:\Costas\Desktop\ODCFSDinfo.exe	9DBFF96D5CF7FDC4CD0C5285D3CB1844	861ED00E54296AAE27AC66F8EAAED103249...

in WinHex, saved it (*obviously the MD5/SHA1 hashes are now different*):

Filename	Full Path	MD5	SHA1
ODCFSDinfo.exe	D:\Costas\Desktop\ODCFSDinfo.exe	3E24338CF560C553A9218975789BBC1B	62B5DB078B370E651CD31A895956D07CAE8...

and executed it:

ETag	Application	PackageIdHash
825030	D:\Costas\Desktop\ODCFSDinfo.exe	00SlgddmGo2acCj6LWmJ61RsdMpKy5UH8/AeED0hoCA=
825026	D:\Costas\Desktop\ODCFSDinfo.exe	00SlgddmGo2acCj6LWmJ61RsdMpKy5UH8/AeED0hoCA=

And the ‘PackageIdHash’ is still the same! So, it appears that ‘PackageIdHash’ is not hashing the contents of the file. Just the full path (including filename).

I highly suspect that ‘PackageIdHash’ is the hash of the info of the application (*see below*) and then is encoded in Base64:

```
Name      : 33322BoxInc.Box
Publisher  : CN=D1F7D9E8-004F-4357-8F99-A820EE1E0308
Architecture : Neutral
ResourceId   : split.scale-400
Version     : 2.1.68.0
PackageFullName : 33322BoxInc.Box_2.1.68.0_neutral_split.scale-400_dbfg57zya1ymg
InstallLocation : C:\Program Files\WindowsApps\33322BoxInc.Box_2.1.68.0_neutral_split.scale-400_dbfg57zya1ymg
IsFramework  : False
PackageFamilyName : 33322BoxInc.Box_dbfg57zya1ymg
PublisherId   : dbfg57zya1ymg
IsResourcePackage : True
IsBundle      : False
IsDevelopmentMode : False
NonRemovable   : False
IsPartiallyStaged : False
SignatureKind  : Store
Status       : Ok
```

such as obtained by

```
Get-AppxPackage 33322BoxInc.Box
```

Or

```
Mode      : -a---
VersionInfo : File: D:\Costas\Desktop\ODCFSDinfo.exe
InternalName:
OriginalFilename: ODC-FSD
FileVersion: 1.0.8.0
FileDescription:
```

```
Product:      ODCFSInfo
ProductVersion: 1.0.8.0
Debug:        False
Patched:      False
PreRelease:   False
PrivateBuild: False
SpecialBuild: False
Language:     Language Neutral

BaseName    : ODCFSInfo
Target      : {}
LinkType    :
Name        : ODCFSInfo.exe
Length      : 124472
DirectoryName : D:\Costas\Desktop
Directory   : D:\Costas\Desktop
IsReadOnly   : False
Exists       : True
FullName    : D:\Costas\Desktop\ODCFSInfo.exe
Extension   : .exe
CreationTime : 07 Jul 20 9:40:58 pm
CreationTimeUtc : 07 Jul 20 6:40:58 pm
LastAccessTime : 26 Jul 20 2:39:26 pm
LastAccessTimeUtc : 26 Jul 20 11:39:26 am
LastWriteTime : 25 Jul 20 9:47:24 pm
LastWriteTimeUtc : 25 Jul 20 6:47:24 pm
Attributes   : Archive
```

Obtained by:

```
(get-item "D:\Costas\Desktop\ODCFSInfo.exe")|select -Property * -ExcludeProperty PS*
```

Not sure how this works with apps like

“Microsoft.Office.Word_8wekyb3d8bbwe!microsoft.word” or “E7CF176E110C211B” (*Mozilla Firefox*) or “OperaSoftware.OperaWebBrowser.1402082470” etc.

To be updated



[CC Attribution 4.0 International \(CC BY 4.0\)](#)