

presented by



Microsoft Sample Code on GitHub and Walkthrough on Firmware Updates to Windows Update (WU)

Spring 2018 UEFI Seminar and Plugfest

March 26-30, 2018

Presented by Bret Barkelew and Keith Kepler

Facilitated by Michael Anderson



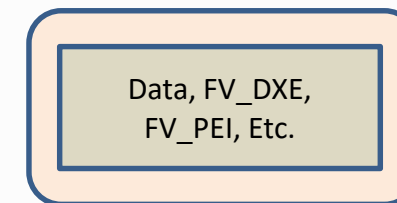
4 Steps to WU (FW Perspective)

- Create the capsule binary payload
- Create the Windows system file (.inf)
- Package and sign WU deliverables (.bin and .inf)
 - If using HLK, this is a .hlkx. Otherwise, use attestation signing on .cab
- Publishing to Windows Update
- **NOTE:** This will be required for 10 in S mode



Growing a Capsule, Inside Out

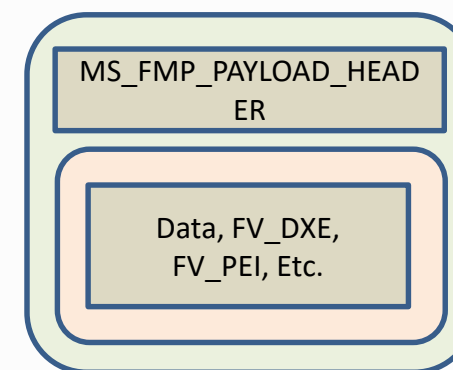
- Start with a payload
- Tools available on GitHub
 - Links at end of section





Growing a Capsule, Inside Out

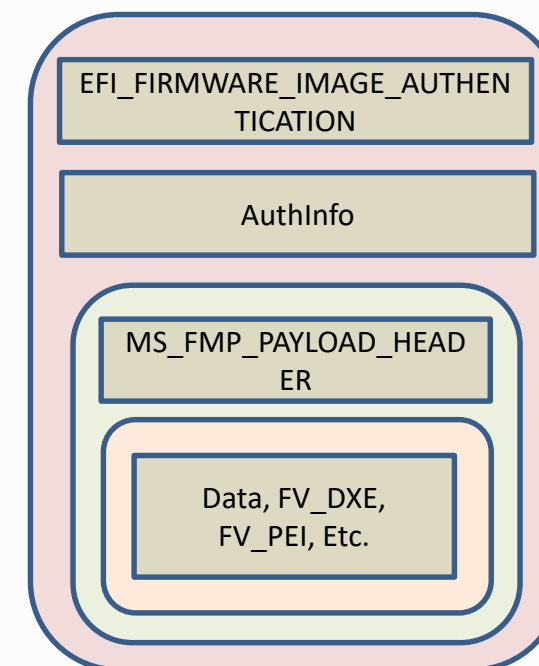
- [Optional Step] Use GenMsPayloadHeader.exe to set Version, LSV, etc.





Growing a Capsule, Inside Out

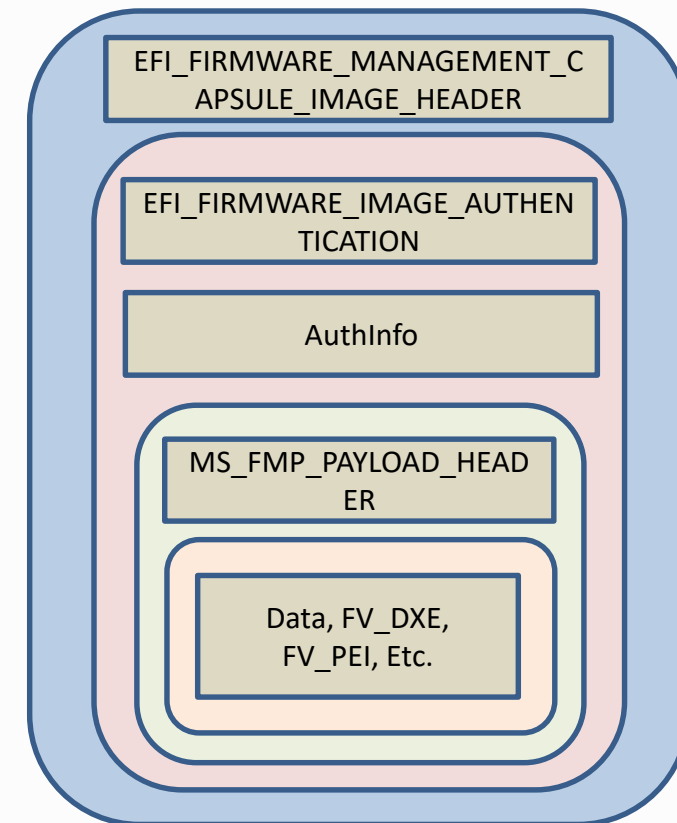
- Use GenFmpImageAuth.py and the capsule driver signing cert to sign capsule
- This signature is for the capsule framework, not Windows. It will be evaluated by the FMP driver





Growing a Capsule, Inside Out

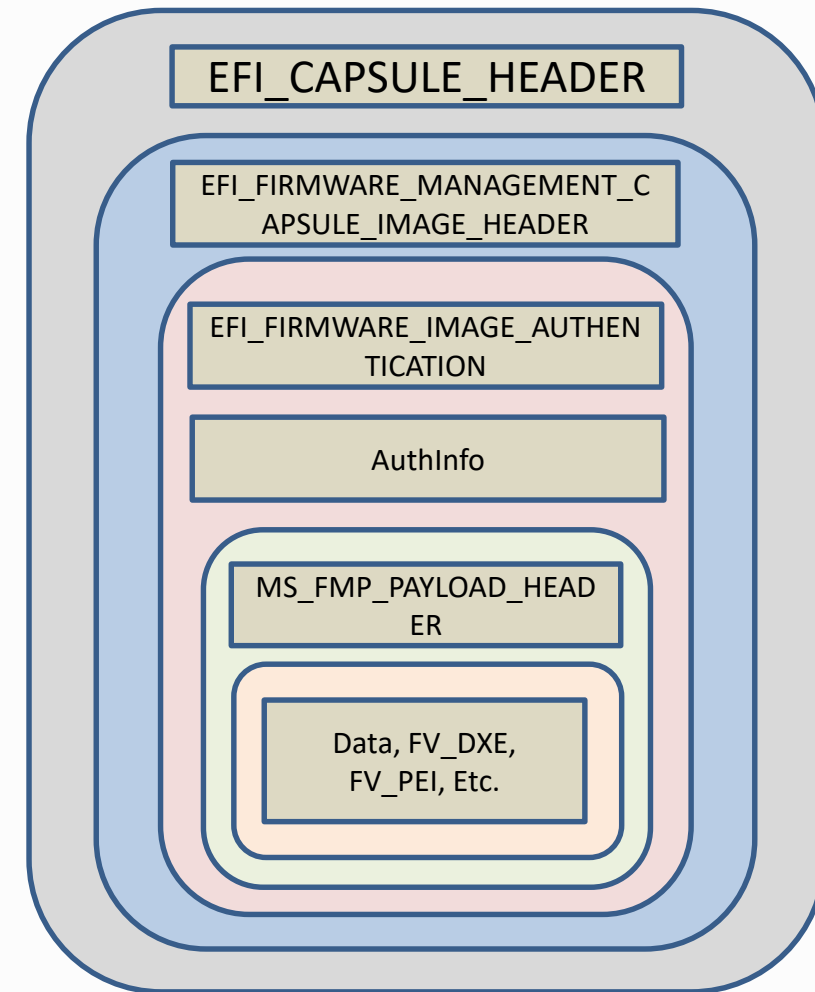
- Use GenFmpCap.exe to wrap the signed payload with a GUID that indicates at which driver it is targeted
- This GUID is the one declared in the ESRT (and .inf file)





Growing a Capsule, Inside Out

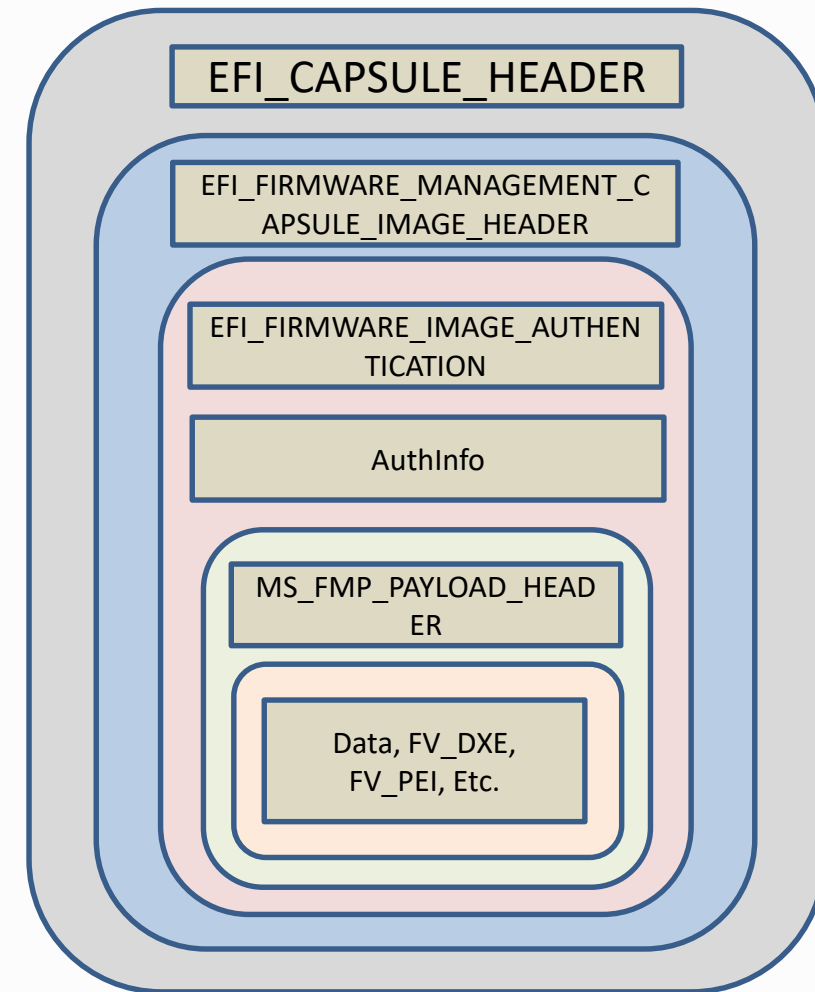
- Use GenFv.exe to wrap the FMP payload in a capsule wrapper
- This wrapper uses the gEfiFmpCapsuleGuid to indicate it should be processed by the framework



Growing a Capsule, Inside Out



- Et voila...
- This binary payload is now ready for WHQL signing (once we pair it with an .inf and .cat file)





To .INFINITY, and beyond

- Use CreateWindowsCapsule.py to generate the Windows .inf file to accompany the binary payload
 - Does not consume the payload, but the name parameter must match
- This utility will also create a .cat file and test-sign it (with .pfx specified in parameters) for validation on test machine [optional for .cab attestation signing]



Tools on GitHub

MS_UEFI/MsSampleFmpDevicePkg/Tools/ - Build Process and Windows Tools

https://github.com/Microsoft/MS_UEFI/tree/share/MsCapsuleSupport/MsSampleFmpDevicePkg/Tools

MS_UEFI/MsSampleFmpDevicePkg/Tools/GenTools/ - Capsule Binary Payload Generation Tools

https://github.com/Microsoft/MS_UEFI/tree/share/MsCapsuleSupport/MsSampleFmpDevicePkg/Tools/GenTools

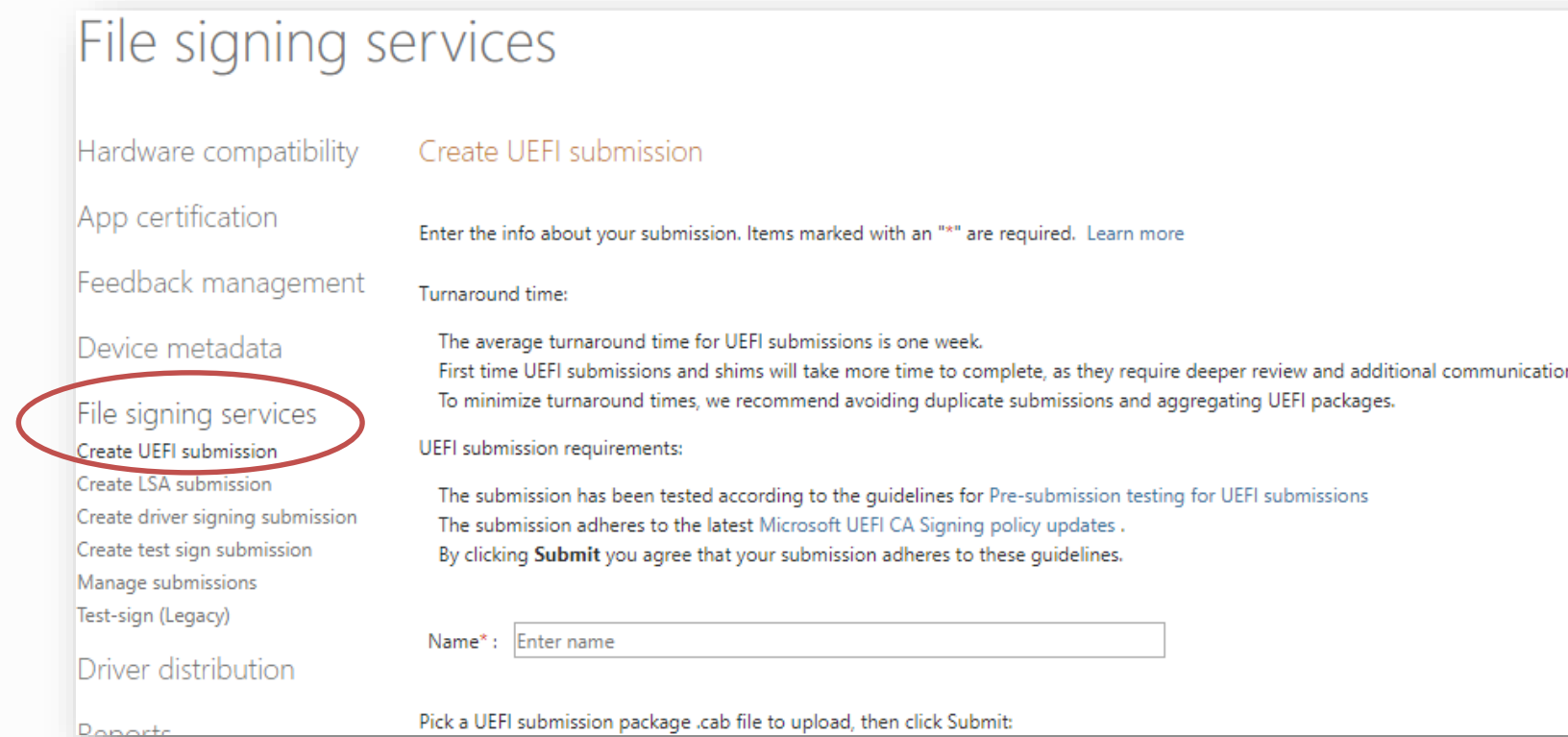
Example: Attestation signing a Windows driver (same process for Capsule .cab file and EV signing)

<https://docs.microsoft.com/en-us/windows-hardware/drivers/dashboard/attestation-signing-a-kernel-driver-for-public-release>

UEFI EFI Signing - Sysdev



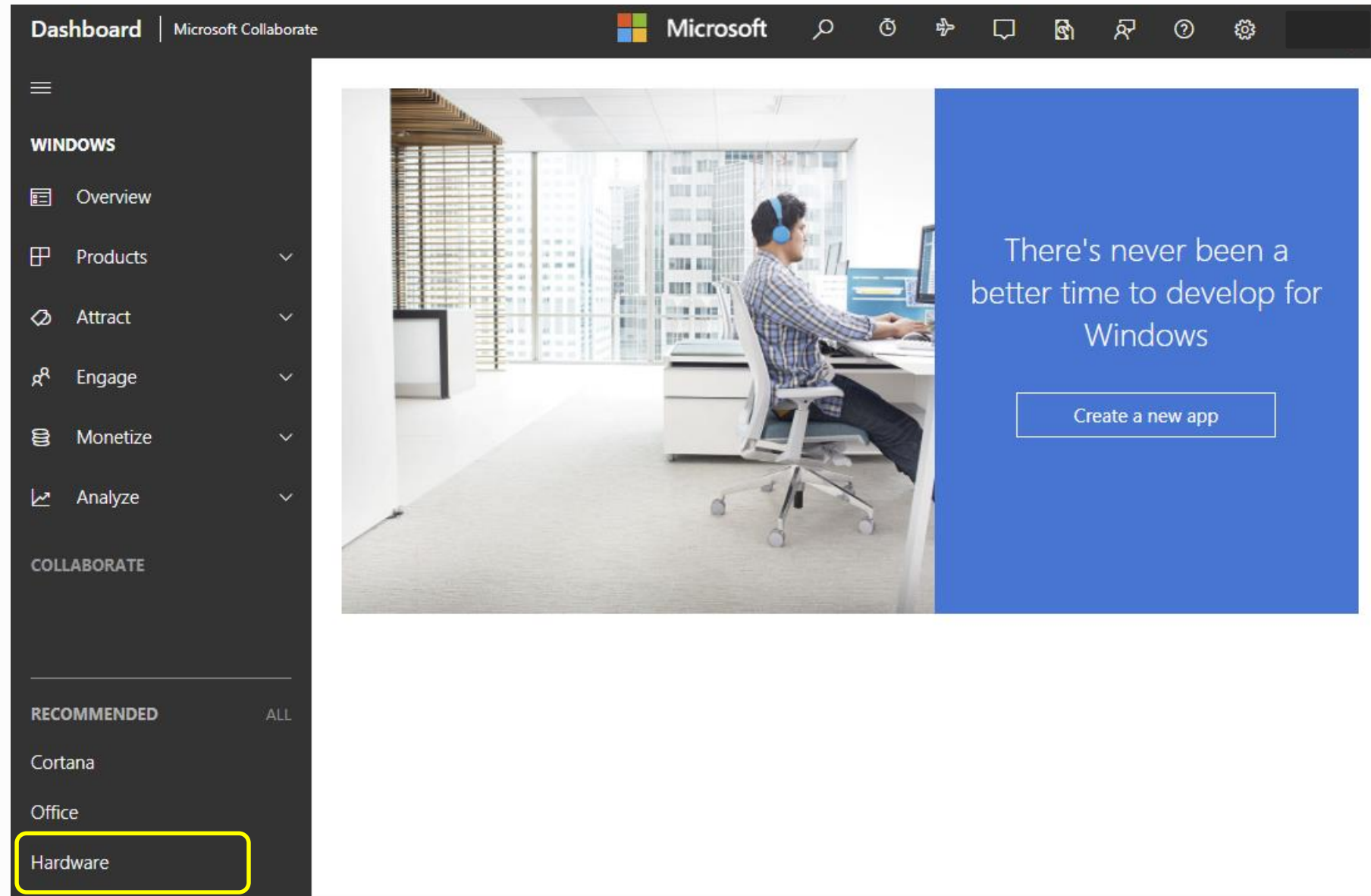
- <https://sysdev.microsoft.com> - [UEFI help section](#)
- Used for signing .EFI files, not for UpdateCapsule firmware updates
- .CAB must be EV signed before submitting to Sysdev



Walk through for Signing UEFI Pkg



- <https://developer.microsoft.com/dashboard>



Submit new hardware

<https://developer.microsoft.com/dashboard/hardware>



The screenshot shows the Microsoft Developer Dashboard for Hardware. The page title is "Hardware" and it includes a "Submit new hardware" button highlighted with a yellow box. Below the button is a table with columns: Shared Product ID, Failure Hits, Name, State, Certification type, Created date, Permission, and Source. The table contains three rows of data.

Shared Product ID	Failure Hits	Name	State	Certification type	Created date ↓	Permission	Source
1152921504607		Test_SubInfo	Completed	WLK	3/23/2018	Author	SpiralOrbit Hardware Dev Center
1152921504607		Test_WLK	Preparation	WLK	3/23/2018	Author	SpiralOrbit Hardware Dev Center
1152921504607		ExtensionINFIntTesting	Completed	HLK	3/23/2018	Author	SpiralOrbit Hardware Dev Center

Submit your HLKx or .CAB capsule package



A screenshot of the Microsoft Developer Dashboard for hardware driver submission. The browser address bar shows 'https://developer.microsoft.com/en-us/dashboard/hardware/driver/New'. The page title is 'New hardware submission'. A progress bar at the top shows eight steps: Package Acceptance (active), Preparation, Scanning, Validation, Catalog creation, Manual review, Sign, and Finalize. Below the progress bar is a section titled 'Packages and signing properties' with a 'Product name*' input field and a large grey area for dragging packages. Below that is a 'Requested Signatures' section with explanatory text. At the bottom are 'Certification' and 'Distribution' sections, both with red lock icons. A left sidebar contains navigation options like 'Drivers', 'Systems', 'IoT', and 'Analyze'. A search bar in the sidebar is highlighted with a yellow box, containing the text 'Test_SubInfo', 'Test_WLK', and 'ExtensionINFlntTesting'. The Microsoft logo is visible in the top right of the dashboard.



Local Testing

- Download your signed capsule update package
- DevMgr->Firmware->Driver->Update Driver or
- Right click the INF and choose *Install*

Package Acceptance Preparation Scanning Validation Catalog creation Sign Finalize

Your submission is complete! Your signed packages are available below. Any shipping labels you created earlier are processing.

Packages and signing properties

	Date	Submission Id	Status	Created By	
FirmwareUpdateINF.cab-SEANQU_DON'T_PROCESS_-63640989720 - Initial	9/14/2017	1152921504627024137	Finalize	Sean Quiriconi	

[Download signed files](#) [Download initial package](#)

Publish to Windows Update



My first driver
ID: 1152921504606955856

1

Package Acceptance Preparation

We detected some issues while validating your package and are man

Packages and signing properties

Upload new Download DUA shell

Name	Date
My first driver - Initial	10/24/2016

This is a beta driver

Additional certifications
No additional certifications were chosen.

Certification

External

Certification form complete

You have completed the certification, but Microsoft needs to process

Distribution

New shipping label Publish all pending

First publication

2

Created Validation UpdateGeneration Publishing Finalize

Details

Shipping label name
First publication

Properties

Destination

Publish to Windows Update

Send to another Partner

Release date

Publish my driver as soon as it passes certification

No sooner than

Specify the partner (if any) that is allowed visibility into this request.

Enter search term

Driver promotions

Automatically deliver and install this driver during Windows Upgrade

Automatically deliver and install this driver on all applicable systems

By default, drivers are only delivered if a device does not already have a driver installed. These options allow you to override the default behavior but require additional Microsoft eva

Targeting

3

Select driver package
My first driver (Initial Package)

You can add new driver packages by uploading an updated driver package in the package section of this driver.

Select PNPs

Publish All

Search by hardware ID Enter search term Please fill out this field. operating system Enter search term

Hardware ID	Operating system	INF	Status	
acpi\jen2021	Windows 7 Client	Apfiltr.inf	Pending publish	Cancel *
acpi\jen2021	Windows 8.1 Client	Apfiltr.inf	Not on Windows Update	Publish *
acpi\jen2021	Windows 10 Anniversary Update Client	Apfiltr.inf	Not on Windows Update	Publish *
acpi\jen2021	Windows 7 Client x64			
acpi\jen2021	Windows 8.1 Client x64			

Select CHIDs

4

Enter new CHIDs

Add CHID(s) Add multiple CHIDs

Windows Insider Program (WIP)



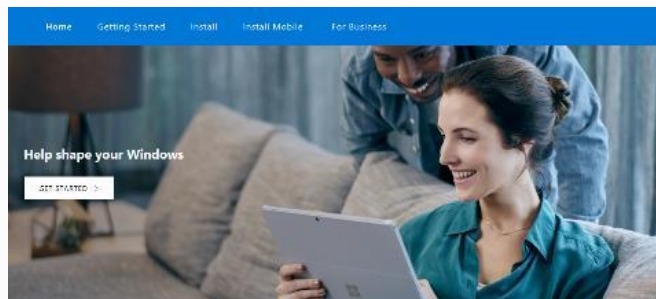
- All Firmware published to WU goes through WIP (can add up to 2 weeks)
- Status is "Microsoft approval"
- Great opportunity for real world testing with systems

Two Steps to become an Insider



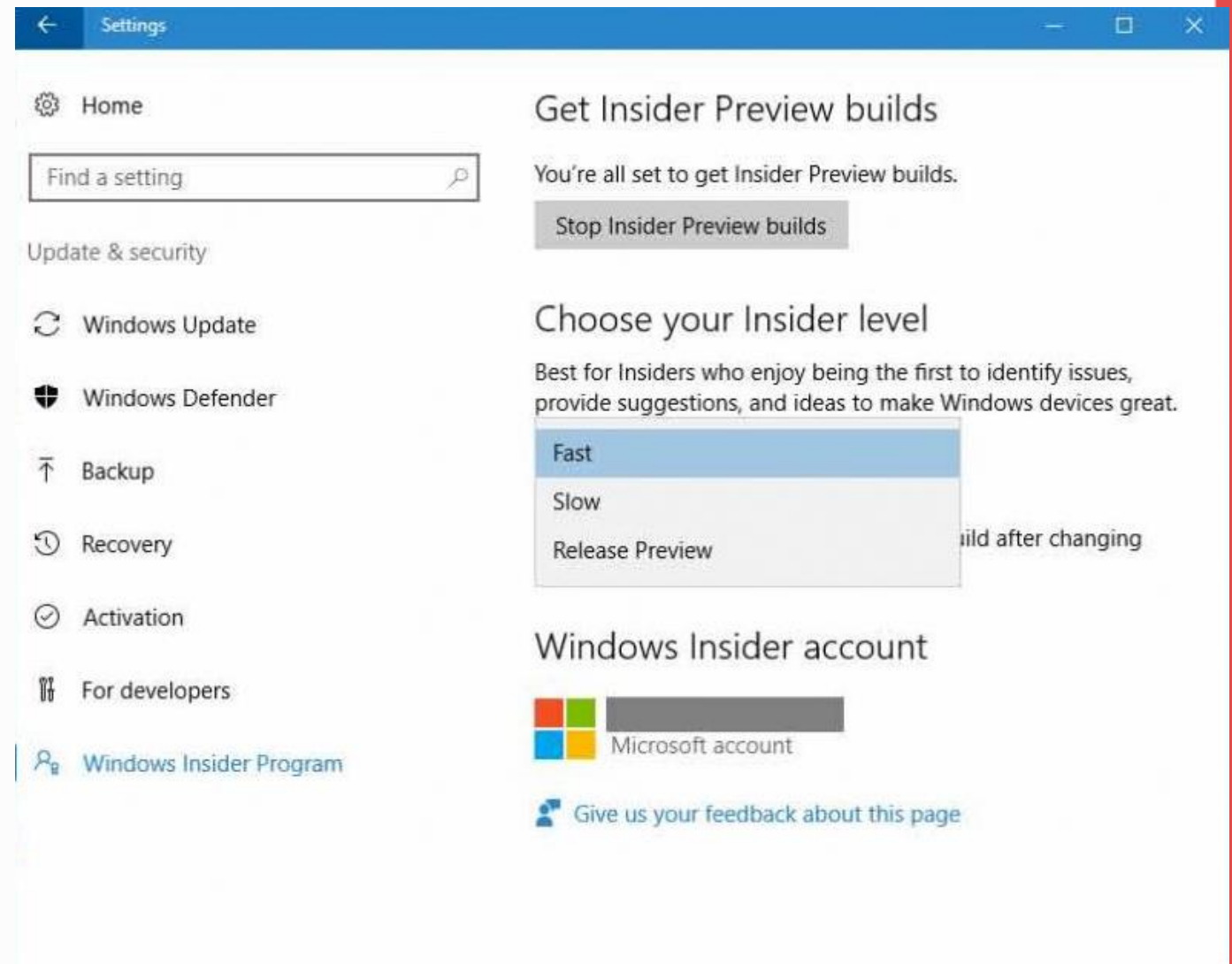
1

Sign up at
<http://insider.windows.com>



2

Update each client as needed

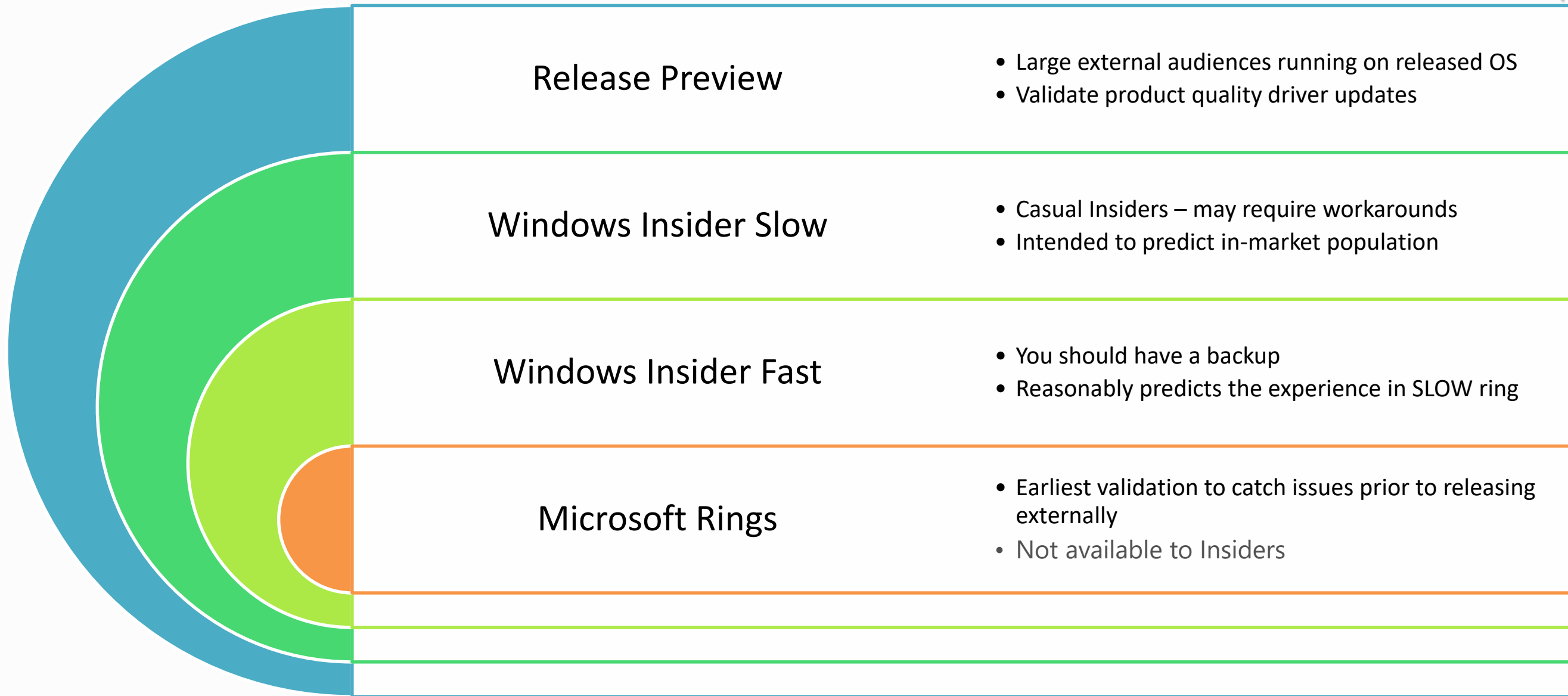


Best Practice

Support both *Released OS* and *Upcoming OS*

Insider Level	OS
Fast or Slow	Upcoming
Release Preview	Released

Each Ring has an Intended Experience



Rings enable the broadest set of customers to engage at their own risk level

Important Terms



Windows Insiders

Fans of the Windows 10 OS who want to experience latest offerings from Microsoft in order to guide and shape future releases of the operating system

Ring

A defined customer experience that guides how drivers are flighted to Windows

Flighting

The automatic distribution, monitoring and evaluation of a driver to a set of customers in defined rings

Shipping Label

The concept in DevCenter used to define the destination of a driver and the method of delivery



Questions?



Thanks for attending the Spring 2018 UEFI Plugfest

For more information on the UEFI Forum and
UEFI Specifications, visit <http://www.uefi.org>

Questions regarding this presentation <mailto:SAUEFI@Microsoft.com>

presented by

