In-Market Automation Execution E2E Transition

Contents

General Introduction

Application Compat team uses automated scripts to assess the OS changes. To test the daily OS changes for compatibility, automated scripts are scheduled, and the results are triaged to derive insights on the compatibility health on a given OS scenario. Compat Sign off library comprises of various apps selected based on comprehensive selection criteria (that includes MAD, app categories, and various app capabilities/complexities) that are already automated and running across various In-Market and Win Next OS scenarios. Currently we are using WTT/ES platform for scheduling AD, CAC Automation runs and MTP Runs

1. App Automated: App is automated using aelib tools into C# scripts.
2. The script code entered Git hub for storing, tracking changes etc.
3. The code needs to be approved by Microsoft project managers.
4. Time build generation: Binaries for each automated app get created. These contain required script, configuration files, languages file etc.
5. Binaries Copied: The generated binary folders must be copied to the source location.
6. Job ID created: App automation runs are executed through jobs, Each app has a unique job id created to execute it
7. Baseline Run 5x: To test how well the app runs on the test platform and resolve any fundamental issues.
8. Tagged to collection: Depending on the type of app each automated app must be tagged to a collected for example: Win32, Modern, INTL, VM Applicable, Phy specific etc…

9. **Run Triggered:** kick start automation runs in the production environment on the targeted test build.
10. **Progress Tracking:** Monitoring the automation execution status through OCAS/WTT/ACG
11. **Run Completed:** Once run is completed, the results will be available in OCAS portal for triaging the failed runs
12. **Triage performed:** All the failed app runs need to be triaged to check the root cause of the failure (Ex: App related issue, Script/Infra related issue)
13. **Bugs uncovered:** Bugs are logged for the regression and non-regression issues.
14. **Bugs Fixed:**
15. **Script issues for fixing team:** Any non-application related failures need to be analyzed further and fixed.
16. **Report Generated:** Once all the failures are completed triaged, the triage report is submitted generating the final run summary report for everyone to view.

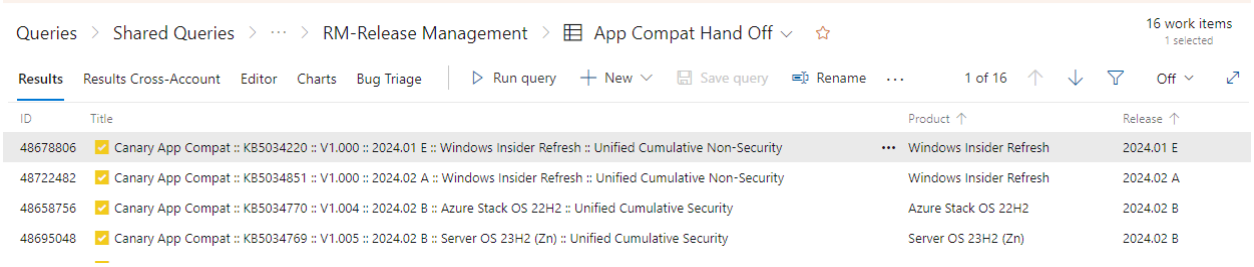
Understanding Track-it Portal, Hand-off Query

This is to understand the priority task that must be executed as part of CAC execution.

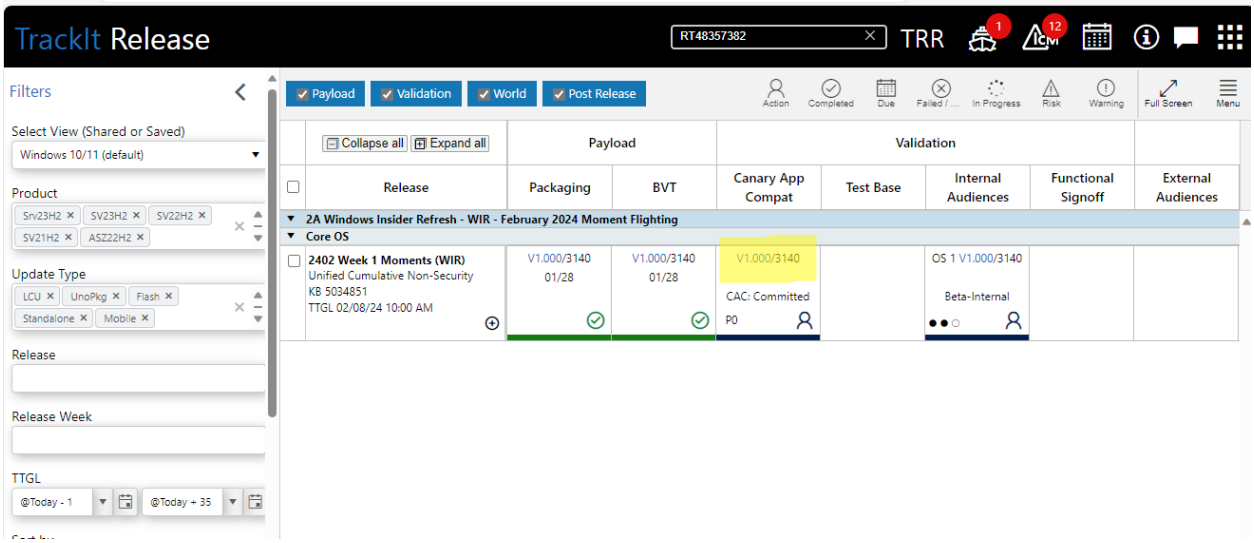
Open the appcompat hand off query Here to list all the tasks available.

- Check for the task available for the latest releases.
- Check the task with the nearest TTGL.
- Check for the priority mentioned, need to pick the task starting with the priority set to 0. If there is no task with priority 0 then move to task with priority 1 and 2.
- Open the task and click the **Trackit Report** link available in the description.
- Check for the package availability and if there is any new version in progress.
- Download the package for the link available in the Trackit in Canary App Compat column with the CAC state indicated as committed. (Highlighted in the snip)
- Snips of the Track it and app compat hand off query.

AppCompat Hand Off



Trackit Report



Note: Can use the Track it to check the availability of any package for specific scenario by selecting the options provided in the Filters pane available on the left side of the page.

AEAUTO Framework, Test binaries & AELIB Tools Introduction

The AEAuto Framework is a set of API's and tools created by the Application Compatibility Gates (ACG) team to facilitate testing Microsoft operating systems within the realm of application compatibility. As such, it is a set of API's and tools which specialize in application automation. This specialization is what differentiates the AEAuto Framework from the overreaching TAEF (Test Authoring and Execution Framework).
Framework Customers:

- People that maintain the framework
- People writing new test automation
- People maintaining existing test automation
- People triaging test execution or consuming results of execution
- People that the framework depends on to operate - 3rd party dependencies / reciprocal customers
- People in management that need to report on gates operation.

Name	Description	Location
AELIB Binaries	Pre-Requisite for running the automated binaries with AE Auto Framework	Hyderabad: \\wsehyd-applib\AutoExecution\Tools\Latest_Aelib\aelib Redmond:
Test Binaries	Binaries generated with automation code placed in git Location	Hyderabad: \\wsehyd-applib\AutoExecution\Hybrid Redmond: \\sesdfs\1windows\ACGATES\Logs\appcompat\auto\tests

Current Automation Test Coverage

Parameter	Canary	MTP	Canary Private
Frequency	Daily	Monthly	On-Demand
SLA	12 Hours	10 Days	Release ETA
Target	4 CACs	14 OS for B cycle and 4 OS for D Cycle	On-Demand
OS Coverage	SV23H2, SV22H2, SV21H2, Windows 10 22H2, RS5, RS1, TH1 and WS2022	**Client:** SV23H2, SV22H2, SV21H2, Windows 10 22H2, RS5, RS1, TH1, Windows 7 **Server:** WS08R2SP1, WS2012R2, WS2016, WS2019, WS2022	SV22H2

Scheduling the Runs

OCAS Execution is leveraging both WTT & ES Platforms for CAC & MTP Full Test Pass efforts. If a product is onboarded on ES, we are ensuring to leverage only the ES Platform for specific Product. In case there are issues with ES Infrastructure (Down time/ Any other Issues) we shall utilize the WTT Platform to proceed with Execution

WTT Platform

Pre-Requisites

WTT Tool and Respective Pool permissions

Test Bed Setup

Available Pools:

1. WTT – Hyderabad ($\AppCompat-Validation\AutoExecution)
2. WTT – ESCPlaceholder ($\AppCompat-Validation\AutoExecution)

Job Details

	JOB Name	Hyderabad	Redmond	Comments
1	SCHEDULE OS	258, 22970 [SV]	32071	To deploy OS
2	POST OS	592	26213	Automation req
	POST OS (Server)	25033	292040	
3	.Net install	430	126167	.Net Prerequisites
4	Live updates & additional	20138	279219	Windows updates
5	Package installation	18122	264369	Test package
	Private Package install	20276	296631	Test package
6	Screen Resolution	20096	NA	Set screen res
7	Enable NTVDM	NA	270442[only x86]	Enable NTVDM
8	Create Image	24987	26061	Capture Image
9	Restore Image	435	231305	Restore Image

Package's location:

Hyderabad:\\wsehyd-applib\ae\Automation\Execution\OfflinePkg

Redmond:

Build Locations:

Product	Arch	Share Location
Windows 11 22H2/23H2/ Moments	X64	Redmond: \\winbuilds\release\ni_release\22621.1.220506-1250\amd64fre\media\enterprise_en-us_vl Hyderabad: \\wsehyd-oslib\MSOS\PreRTM\NI_RELEASE\22621.1.220506-1250\amd64fre\media\enterprise_en-us_vl
Windows 11 21H2	X64	Redmond: \\winbuilds\release\co_release\22000.1.210604-1628\amd64fre\media\enterprise_en-us_vl Hyderabad: \\wsehyd-oslib\MSOS\PreRTM\CO_RELEASE\22000.1.210604-1628\amd64fre\media\enterprise_en-us_vl
Windows 10 22H2	X64	Redmond: \\winbuilds\release\vb_release\19041.1.191206-1406\amd64fre\media\enterprise_en-us_vl Hyderabad: \\wsehyd-oslib\MSOS\PreRTM\VB_RELEASE\19041.1.191206-1406\amd64fre\media\enterprise_en-us_vl
	X86	Redmond: \\winbuilds\release\vb_release\19041.1.191206-1406\x86fre\media\enterprise_en-us_vl Hyderabad: \\wsehyd-oslib\MSOS\PreRTM\VB_RELEASE\19041.1.191206-1406\x86fre\media\enterprise_en-us_vl
Windows 10 1809(RS5)	X64	Redmond: \\winbuilds\release\rs5_release\17763.1.180914-1434\amd64fre\media\client_ltsc_en-us_vl Hyderabad: \\wsehyd-oslib\msos\PreRTM\RS5_RELEASE\17763.1.180914-1434\amd64fre\media\client_ltsc_en-us_vl **SKU:** ENTERPRISES
	X86	Redmond: \\winbuilds\release\rs5_release\17763.1.180914-1434\x86fre\media\client_ltsc_en-us_vl Hyderabad: \\wsehyd-oslib\msos\PreRTM\RS5_RELEASE\17763.1.180914-1434\x86fre\media\client_ltsc_en-us_vl **SKU:** ENTERPRISES
Windows 10 1607(RS1)	X64	Redmond: \\winbuilds\release\rs1_release\14393.0.160715-1616\amd64fre\enterpriseS_en-us_vl Hyderabad: \\wsehyd-oslib\MSOS\Released\RS1_RELEASE\14393.0.160715-1616\amd64fre\media\enterpriseS_en-us_vl

		SKU: ENTERPRISES
	X86	Redmond: \\winbuilds\release\rs1_release\14393.0.160715-1616\x86fre\enterpriseS_en-us_vl Hyderabad: \\wsehyd-oslib\MSOS\Released\RS1_RELEASE\14393.0.160715-1616\x86fre\media\enterpriseS_en-us_vl **SKU:** ENTERPRISES
Windows 10 RTM (TH1)	X64	Redmond: \\winbuilds\release\th1\10240.16384.150709-1700\amd64fre\enterpriseS_en-us_vl Hyderabad: \\wsehyd-oslib\msos\RELEASED\TH1\10240.16384.150709-1700\amd64fre\media\enterprises_en-us_vl **SKU:** ENTERPRISES
	X86	Redmond: \\winbuilds\release\th1\10240.16384.150709-1700\x86fre\enterpriseS_en-us_vl Hyderabad: \\wsehyd-oslib\msos\RELEASED\TH1\10240.16384.150709-1700\x86fre\media\enterprises_en-us_vl **SKU:** ENTERPRISES
Windows 7	X64	\\wsehyd-applib\AE\Public\Image\Win7\effigyimage\x64\7601.26910.amd64fre.win7_release_svc_refresh.240109-1401_client_enterprise_en-us_vl.WIM
	X86	\\wsehyd-applib\AE\Public\Image\Win7\effigyimage\x86\7601.26910.x86fre.win7_release_svc_refresh.240109-1401_client_enterprise_en-us_vl.WIM
Windows Server 2022	X64	Redmond: \\winbuilds\release\fe_release\20348.1.210507-1500\amd64fre\media\server_en-us Hyderabad: \\wsehyd-oslib\MSOS\PreRTM\FE_RELEASE\20348.1.210507-1500\amd64fre\media\server_en-us **SKU**: serverstandard
Windows Server 2019(RS5 Server)	X64	Redmond: \\winbuilds\release\RS5_RELEASE\17763.1.180914-1434\amd64fre\media\server_en-us Hyderabad: \\wsehyd-oslib\MSOS\PreRTM\RS5_RELEASE\17763.1.180914-1434\amd64fre\media\server_en-us **SKU:** serverstandard
Windows Server 2016(RS1 Server)	X64	Redmond \\winbuilds\release\RS1_RELEASE\14393.0.160715-1616\amd64fre\media\server_en-us Hyderabad:

		\\wsehyd-oslib\msos\Released\RS1\amd64fre\media\server_en-us\14393.0.160715-1616 **SKU:** serverstandard
Windows Server 2012 R2 (Win8.1 Server)	X64	Hyderabad: \\wsehyd-ode\ODE_Images\Win8.1\effigyimage\Server\en.2k12r2.serverstandard.rtm.fpp.Patched.wim
Windows 2008 R2 Sp1 (Win7 Server)	X64	Hyderabad: \\wsehyd-ode\ODE_Images\Win7\effigyimages\Server\wim_serverstandard_en-us.wim

Machines preparation Process:

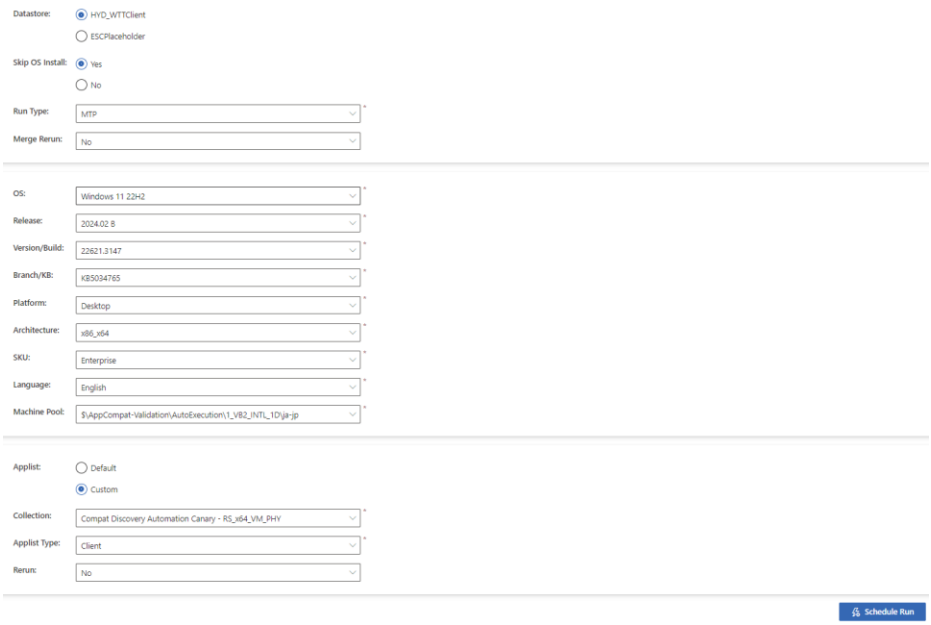
- Create a separate Pool for Machine Preparation
- Move the available Machines to Specific Pool.
- Start Running all the jobs(1 to 7) required and ensure there are no failures observed.
- Observer the expected winver (WTT\OSBuildNumber, WTT\SPBuildNumber)
- Run Create Image Job
- Move the machines now to new Pool (create it for Triggering Runs)

Schedule trigger

Once the test Bed Setup is Over, we need to use the OCAS portal to trigger the Runs.

Steps for Triggering the Runs:

- Navigate to Create Schedule page from OCAS. Alternatively Click the link OCAS :: OS Compat AutoSuite (msappproxy.net)
- Select the Respective Datastore Checkbox where Machine preparation Done.
- Select the Yes Checkbox for Skip OS Install
- Select the Run type as Canary/MTP
- Select the OS for which Automation Run needs to be Triggered.
- Select the Respective Release, Version, KB Details as shown below.
- Select the platform as Desktop.
- Select the Respective Arch(x64/x86/x86_x64)
- Select applicable SKU based on the OS
- Select the Language as English for normal Runs and Respective Language for INTL Runs
- Select the Respective Pools where Machines are placed for Execution Run.
- Select Custom Checkbox under Applist
- Select the Respective Collections where the Required applications are available.
- Select the Respective Applist Type
- Keep the Re-Run option as NO
- Click on Schedule Run



Monitoring the Run Progress

- Launch WTT and Navigate to Explorer->Result Collection
- Respective Runs will be triggered.
- We can monitor the runs within 2 Hours' Time-gap until completion.

ES Platform

Portal Link: Live : ACG-Maestro

Trigger API: http://acg-maestro/api/schedule/trigger/XXXX

ES Schedules

Canary

OS Type	OS	Test Pass Type	
		VM	Physical
Client	RS_PRERELEASE	1276	1277
	Ni_Prerelease	1443	1444
	Moments (EKB + Staging)(ON)	1272	1274
	Moments (EKB + Staging)(OFF)	1737	1738
	Win 11 23H2(EKB)	1640	1641
	Win 11 22H2 (SV22H2)	1278	1279
	Win SV 21H2 (SV21H2)	1280	1281
	Win 10 22H2 (22H2)	1441	1442
	RS5	1680	1550
	RS1	1505	1579

MTP

	OS	Test Pass Type			
		Win32	Modern	Physical	Games
Client	Moments	1204	1205	1206	1711
	SV 23H2	1650	1651	1652	1696
	SV 22H2	1198	1199	1200	1681
	SV 21H2	1163	1164	1165	1682
	VB 22H2	1439	1427	1440	1683
	RS5	1690	-	1692	1691
	RS1	1693	-	1695	1694

INTL

Language	WinNext	SV23H2	Moments	NI_RELEASE	CO_RELEASE	RS5	RS1
RU-RU (Russian)	1463	1667	1466	1467	1468	-	-
FI-FI (Finnish)	1455	1654	1461	1460	1459	-	-
FR-FR (French)	1445	1668	1474	1475	1476	-	-
NB-NO (Norwegian)	1407	1665	1480	1485	1486	-	-
ES-ES (Spanish)	1447	1669	1496	1497	1498	1697	-
ES-MX (Mexican)	1408	1664	1510	1511	1512	-	-
PT-BR (Portuguese)	1446	1670	1518	1519	1520	-	-
DA-DK (Danish)	1522	1663	1525	1526	1527	-	-
NL-NL (Dutch)	1448	1662	1531	1532	1533	-	-
KO-KR (Korean)	1535	1660	1538	1539	1540	-	-
ZH-TW (Chinese Traditional)	1542	1661	1546	1547	1548	-	-
SV-SE (Swedish)	1551	1659	1554	1555	1556	-	-
PL-PL (Polish)	1558	1658	1561	1562	1563	-	-
IT-IT (Italian)	1565	1657	1568	1569	1570	-	-
HE-IL (Hebrew)	1572	1656	1575	1576	1577	-	-
AR-SA (Arabic)	1587	1666	1590	1591	1592	-	-
ZH-CN (Chinese Simplified)	1604	1671	1607	1608	1609	-	1712
JA-JP (Japanese)	1594	1672	1597	1598	1599	1698	-
DE-DE (German)	1580	1673	1583	1584	1585	-	-

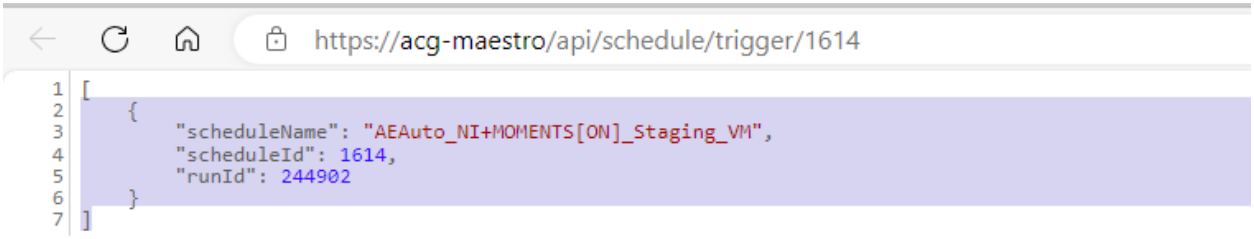
Schedule Trigger

- Replace the Packages in Respective OS Locations as mentioned below.

OS	Package Location	EKB/SSU Location
SV 22H2	\\sesdfs.corp.microsoft.com\1windows\ACGATES\Logs\privateruns\LCUs_Win11Servicing_sv2_22h2_ni_release	NA

SV23H2	\\sesdfs.corp.microsoft.com\1windows\ACGATES\Logs\privateruns\LCUs_Win11Servicing_Moment2\Moments2	\\sesdfs.corp.microsoft.com\1windows\ACGATES\Logs\privateruns\LCUs_Win11Servicing_Moment2\EKB
Feb Moments	\\sesdfs.corp.microsoft.com\1windows\ACGATES\Logs\privateruns\LCUs_Win11Servicing_Moment1\Moments1	\\sesdfs.corp.microsoft.com\1windows\ACGATES\Logs\privateruns\LCUs_Win11Servicing_Moment1\EKB
VB22H2	\\sesdfs.corp.microsoft.com\1windows\ACGATES\Logs\privateruns\LCUs_Win10Servicing_22H2\EKB	\\sesdfs.corp.microsoft.com\1windows\ACGATES\Logs\privateruns\LCUs_Win10Servicing_22H2
RS5	\\sesdfs.corp.microsoft.com\1windows\ACGATES\Logs\privateruns\LCUs_Win10Servicing_rs5\Moments2, \\sesdfs.corp.microsoft.com\1windows\ACGATES\Logs\privateruns\LCUs_Win10Servicing_rs5	\\sesdfs.corp.microsoft.com\1windows\ACGATES\Logs\privateruns\LCUs_Win10Servicing_rs5\EKB
RS1	\\sesdfs.corp.microsoft.com\1windows\ACGATES\Logs\privateruns\LCUs_Win10Servicing_rs1\Moments2, r	\\sesdfs.corp.microsoft.com\1windows\ACGATES\Logs\privateruns\LCUs_Win10Servicing_rs1\EKB

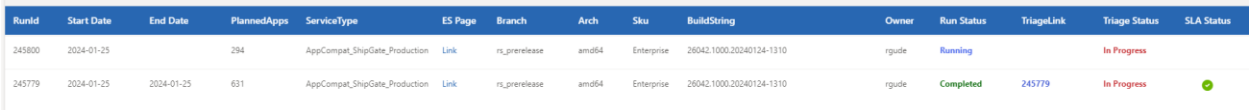
- Use The Trigger API Link and Replace the respective Schedule ID
- We will receive a response from Web Link as below.



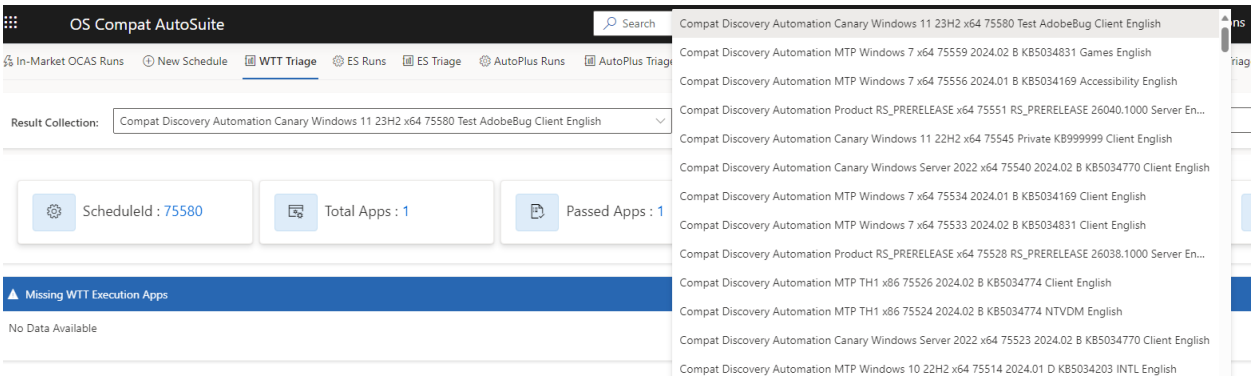
- Note Down the Run ID to search in OCAS Runs Page.
- Monitor the Run Progress from ACG Portal Live : ACG-Maestro

Triage Readiness of OCAS Run

- Once the Runs are completed from Respective Platforms, navigate to OCAS Portal and Pick the Run under ES Runs Tab by manual Search of Respective Run ID
- For ES Run, Click on the Hyperlink available in Triage Link Column as below.



- For WTT Run, Navigate to WTT triage Page, and check the Respective Run ID is listed as below.



- Check the Total apps Count matches the expected Count.

Detailed Triage Process Followed

Manual Re-Run of Binaries

For Any Failures, please perform Re-Run of binaries from respective share location on On-Hold Triage VM's/Physical Machines and observe if the runs are getting passed to ensure there is no underlying regressions.

Generic Guidance on Triaging Failures that failed on Re-Run

Ensure checking only the logs for determining issue Type Since Binaries Re-Run behavior is to confirm the consistency of the issue.

Issue Types	Description
Auto Update	App Controls changing due to application auto update
OS Changes	When a design change included with the new OS build or package or EKB causes the execution failure.
New Popup	Observing new popups while execution. Example: festival offer, update window etc.
Control Change	When there is no app version update, but app behavior change is observed (Vendor site changes, exe location change, app name change etc..
App Services	App Control changes due to change in underlying third-party service/server downtime
Time Out Issues	The app execution halts in the middle to TC execution due to crossing the script allocated execution time.
Credential issues	Credential given by script are incorrect or reached max devices or Trial expired etc..
Setup/Machine issues	Issues that are repro due to slow machine, internet speed & lack of drivers etc. Failure due to typo in the searched path for data files or missing setup (exe) in the given location in script.
Script issue	App execution fails due to script being unable to navigate through the tcs, tagged only when issue is not related to any of the below types.
OS Constraint	The app being executed is not supported on the currently triaged scenario. Due to unsupported OS or architecture.

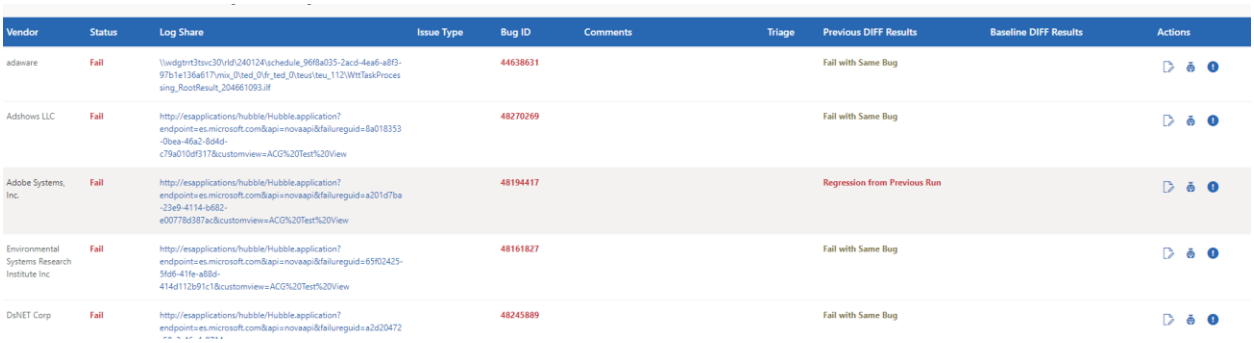
App issue	Any Failure a part of regression, non-regression or Script Issue like server connection failure can be logged before in app health with Bug ID
Known Regression	Regression bug already logged / reported on previous OS build or version package
New Regression	A bug newly logged on latest build or package

Note: please refer Failure analysis PPT for more guidance and tips

Utilizing OCAS Diff V1 and Diff V2 Results for failure Prioritization

OCAS Triage Page Provide the previous Run Diff Results to focus on failures specific to the Run.

Previous Diff Results will be automatically populating in Triage page as below.

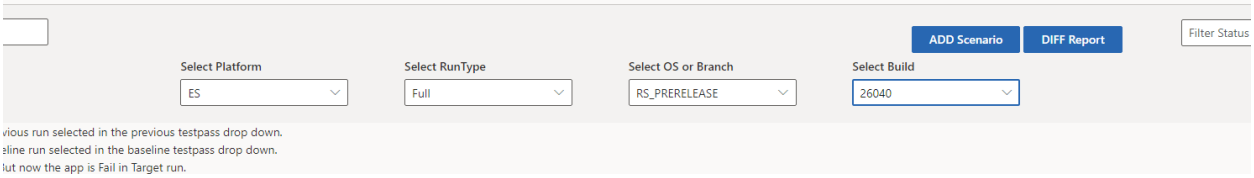


- Failures which show Regression from Previous Run should be prioritized and triaged since those are new failures for the specific run.
- Failures which show failures with the same bug can be tagged with the same Triage category and comments can be updated as same.

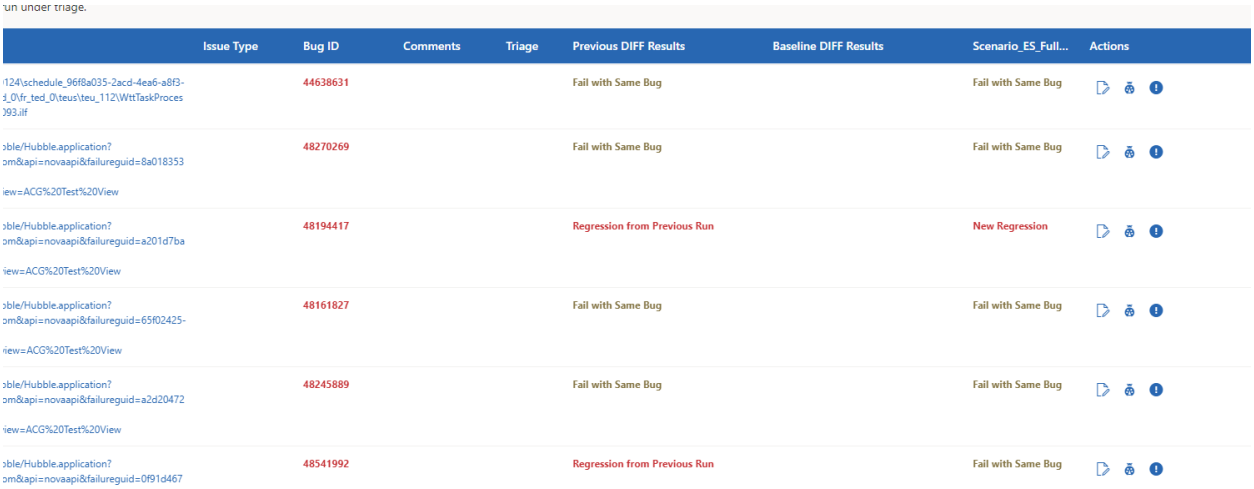
Additionally, we do have Diff V2 where you can compare it with any of the recent reports carried out. Since AD Runs will be weekly, we can select these Run to find the Diff Outcomes

Steps For using Diff V2

Select the Required parameters as Below and click on Diff Report Button

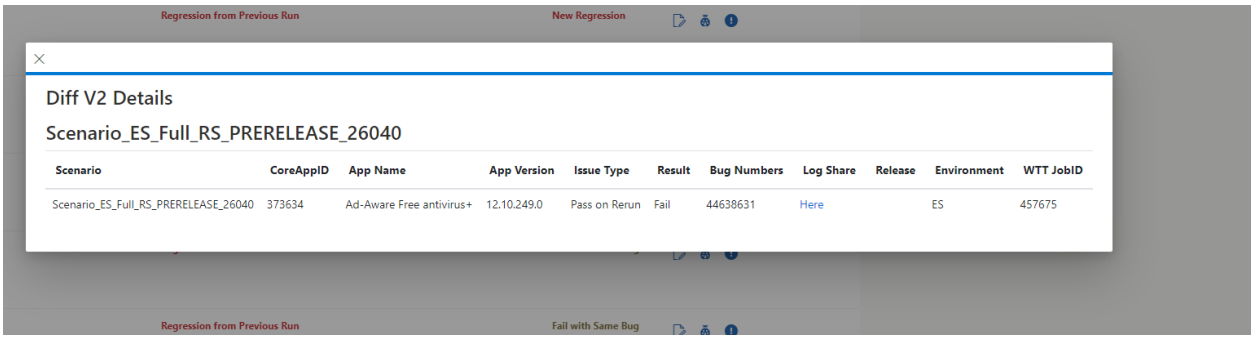


You will be able to see the results below.



Issue Type	Bug ID	Comments	Triage	Previous DIFF Results	Baseline DIFF Results	Scenario_ES_Full...	Actions
\124\schedule_96f8a035-2acd-4ea6-a8f3- d_0\fr_ted_0\teus\teu_112\WttTaskProces 093.ilf	44638631			Fail with Same Bug		Fail with Same Bug	
ble/Hubble.application? m&api=novaapi&failureguid=8a018353 iew=ACG%20Test%20View	48270269			Fail with Same Bug		Fail with Same Bug	
ble/Hubble.application? m&api=novaapi&failureguid=a201d7ba iew=ACG%20Test%20View	48194417			Regression from Previous Run		New Regression	
ble/Hubble.application? m&api=novaapi&failureguid=65f02425- iew=ACG%20Test%20View	48161827			Fail with Same Bug		Fail with Same Bug	
ble/Hubble.application? m&api=novaapi&failureguid=a2d20472 iew=ACG%20Test%20View	48245889			Fail with Same Bug		Fail with Same Bug	
ble/Hubble.application? m&api=novaapi&failureguid=0f91d467	48541992			Regression from Previous Run		Fail with Same Bug	

Also Clicking on “i” will give you the issue Type, Logs and other Details for detailed comparison



Filing In-Market Regressions and Non-Regressions

Regression - Bug which shows unusual /different behavior with pkg, it is known as regression bug. It is a bug that occurs when a previously working feature of the application stops functioning as intended after changes are made to the code, that is, in, Latest Cumulative Update.

Verification Process for Regression Bug - If we find any issue, we first need to verify the issue without 2B package, that is, till windows update, (once regression is confirmed, we need to verify in other lower version of 2B pkg as well if issue is found in higher version). If we found regression issue in any scenario, suppose RS1, we need to verify in higher client OS as well, that is, SV2, SV1, VB2 and TH1. Below is the discussion table for reference:

Sample Discussion Table for Regression Bug:

Below is the sample discussion table for the issue that is found in RS5 scenario with 8B V1.006 pkg.

Verified App Details

App ID	App Name	Version	Vendor	Vendor Site	Reproducibility
298585	Crossfire	3.9.9.0	Tencent	Link	Repro

OS	Scenario	Arch	Build	Reproducibility
RS5	RS5 + WU + UNO(8B KB5029247 v1.006) + App	X64	17763.4733	Repro
RS5	RS5 + WU + UNO(8B KB5029247 v1.005) + App	X64	17763.4730	Repro
RS5	RS5 + WU + UNO(8B KB5029247 v1.006) + App	X86	17763.4733	Not Repro
RS5	RS5 + WU + App	X64/x86	17763.4645	Not Repro
SV2	SV2 + WU + UNO(8B KB5029263 v1.004) + App	X64	22621.2134	Not Repro
SV1	SV21H2 + WU + UNO(8B KB5029253 v1.012) + App	X64	22000.2295	Not Repro
VB22H2	VB22H2 + WU + UNO(8B KB5029244 v1.004) + App	X64/x86	19045.3324	Not Repro
RS1	RS1 + WU + UNO(8B KB5029242 v1.007) + App	X64/x86	14393.6165	Not Repro
TH1	TH1 + WU + UNO(8B KB5029259 v1.005) + App	X64/x86	10240.20105	Not Repro

Non-Regression Bug: A bug that is repro without package too is called as non-regression bug. It is a software bug that occurs independently of any recent changes made to the code.

Verification for Non-Regression Bug: If any issue is found in app, and it is repro without package, then it is a non-regression bug. (Note: Before logging non-regression bug, issue need to be validated in different good configuration machine for the surety). If an issue is found in any OS, it is mandatory to check in higher OS: SV2, SV1 and VB2.

Below is the sample discussion table for non-regression bug:

App ID	App Name	Version	Vendor	Reproducibility
298585	Crossfire	3.9.9.0	Tencent	**Repro**

OS	Clean Scenario	Arch	Build	Reproducibility
RS1	RS1 + WU + APP	X64	14393.6452	**Repro**
SV2	SV22H2 + WU + APP	X64	22621.2715	**Not Repro**
SV1	SV21H2 + WU + APP	X64	22000.2600	**Not Repro**
VB	VB22H2 + WU + App	X64	19045.3693	**Not Repro**

Does this problem continue to happen	Repro/Not Repro
Newer Version of the app/Application Updates	**Repro (v4.0.2)**
Running elevated/Built-in Administrator	NA
With Version Lie (Win7/Win8 Compatibility Mode)	

Additional References

- OCAS_In-Market_Automation_Execution_Support_Guide.docx
- Compat_OCAS_Transition.docx
- ES Schedule Links.docx
- Failure Analysis - OCAS_FailureAnalysis.pptx
- Bug Logging Procedure - Regression and Non-Regression