



# Application Testing & Migration with Rimo3

By Toby Skerritt

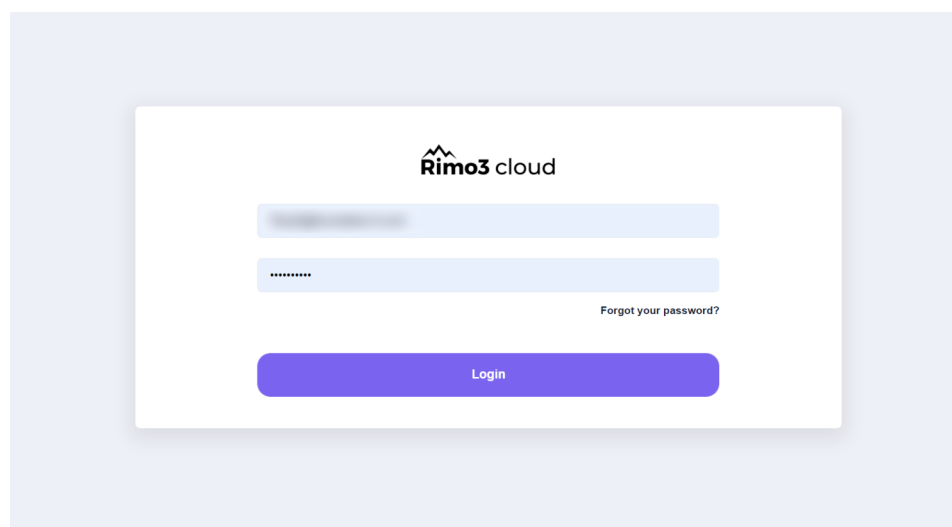
I've had the chance to play with Rimo3's cloud based app modernisation suite and I must say, it provides an excellent alternative to manual application compatibility testing and packaging. For any organisations who use multiple LOB applications, taking Rimo3 for a test drive should be high on the agenda. A lot of this content has already been covered in Dean Cefola's excellent Azure Academy YouTube series, which is definitely worth a look.

<https://www.youtube.com/c/AzureAcademy>

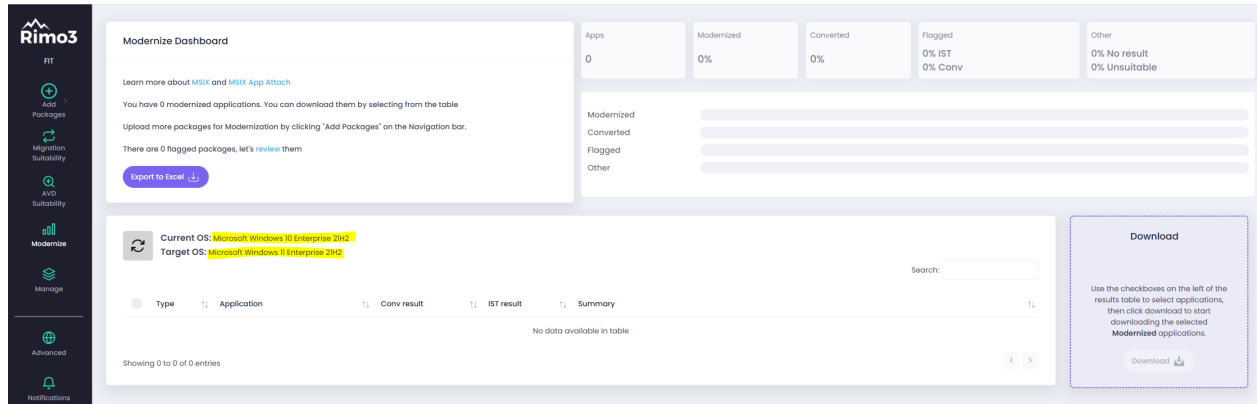
## Weigh Your Options

Rimo3 offers both a cloud-hosted SaaS version, and a self-hosted PaaS offering. The former reduces complexity and allows you to test and validate application performance in a Rimo3 hosted sandbox. The latter allows you to plug the Rimo3 product into your Azure environment, allowing your test VMs or 'Task Runners' with full access to your network, domain and application services. This would obviously be beneficial for domain-integrated or multi-tier applications.

For my testing I was keen to get moving as quickly as possible, so I opted for the Rimo3 SaaS offering hosted at rimo3cloud.com



After logon, you are presented with your Rimo3 Modernise Dashboard, which contains the details for all tested apps, your current OS platform and your target OS platform for validation. It's important to mention here that while the Rimo3 platform provides a wide range of packaging features, a core use case is for ongoing application compatibility testing. If you are planning to roll out a feature update to Windows 10 for example, you can use Rimo3 to confirm application compatibility in the sandbox prior to releasing the update to users.



Rimo3's dashboard is broadly divided into 3 areas:

- Migration Suitability – Will this app work as required on newer versions of Windows?
- AVD Suitability - Will this app work as required in Azure Virtual Desktop, including Multisession environments
- Modernise – convert the package to .MSIX format, and then validate functionality.

I chose to upload and test .MSI files, as the Rimo3 engine is able to automatically identify the name and installation parameters. This would also be true for App-V packages. I could also use executable files, however I would need to specify the installation details manually.

I chose 2 flavours of the Microsoft Azure Virtual Desktop client (x64 and ARM). Obviously, I expect the ARM version to fail testing. Once imported, there is no obvious sign that the applications are being processed, however the progress can be seen under the Advanced > Sequences menu.

Name	Status	Duration	Target OS Image	Test Case(s)
Import - RemoteDesktop_1.2.3004.0_ARM64.zip - 13695	Running	Running	Microsoft Windows 10 Enterprise 21H2 (21H2)	1
Import - RemoteDesktop_1.2.3004.0_x64.zip - 13694	Running	Running	Microsoft Windows 10 Enterprise 21H2 (21H2)	1

Once the apps have been fully imported, their status can be seen on the main console...


 **Current OS:** Microsoft Windows 10 Enterprise 21H2  
**Target OS:** Microsoft Windows 11 Enterprise 21H2

Result of the conversion step

Type	Application	Conv result	IST result	Summary
MSI	Remote Desktop	In Discovery	-	Discovery in progress
MSI	Remote Desktop	In Discovery	-	Discovery in progress

The apps are first validated on the current OS platform, and the re-tested on the Target OS version to confirm compatibility. The apps are then automatically tested against the full range of Rimo3's migration / AVD / modernisation workloads. Essentially, this is a 'fire and forget' process – upload your packages and walk away, then return later and review the output.

Once complete, the Migration Suitability dashboard will give you an overview of the results. As you can see, the x64 package passed both migration suitability and MSIX conversion tests, whereas the ARM package unsurprisingly failed the initial installation task.

 **Current OS:** Microsoft Windows 10 Enterprise 21H2  
**Target OS:** Microsoft Windows 11 Enterprise 21H2

Search:

Type	Application	IST result	Performance	Summary
MSIX	Remote-Desktop	Passed	The Same	This application has passed all tests on the Target OS. Click on the result for more details.
MSI	Remote Desktop	Passed	The Same	This application has passed all tests on the Target OS. Click on the result for more details.
MSI	Remote Desktop	Failed	-	The application failed to install therefore Discovery could not be completed, click on result for more details

We can investigate this failure with the built in tools, including both a video recording and console output. In the case of the ARM package, we can see the dreaded 1603 'fatal error' message during the install process. No surprise here.

```

Console Video Output Performance
2022-04-26T11:05:35:GATEWAY: Agent Task Started: [Install Remote Desktop]
2022-04-26T11:05:37:TASKRUN: Start Task [Install Remote Desktop]
2022-04-26T11:05:37:TASKRUN: Running Pre-Install Scan
2022-04-26T11:05:40:TASKRUN: Pre-Install Scan complete
2022-04-26T11:05:40:TASKRUN: Installing MSI package. Location: T:\2b1b99ff-a159-43b5-b959-82b607d890d7, Install command: msiexec /i "RemoteDesktop_1.2.3004_0_ARM64.msi" /qn
2022-04-26T11:05:42:TASKRUN: Installation from UNC Path failed. Process exited with code: Installation process exited with code 1603
2022-04-26T11:05:42:TASKRUN: Fallback 1: Map UNC Path to a local drive and install
2022-04-26T11:05:42:TASKRUN: Installation from local mapped drive failed. Process exited with code: Object reference not set to an instance of an object.
2022-04-26T11:05:42:TASKRUN: Fallback 2: Copy package locally and install
2022-04-26T11:05:44:TASKRUN: Waiting 10 seconds for installation to complete...
2022-04-26T11:05:54:TASKRUN: Running Post-Install Scan...
2022-04-26T11:05:55:TASKRUN: Scanning the following locations for Executables:
2022-04-26T11:05:55:TASKRUN: C:\Program Files, 0 found
2022-04-26T11:05:55:TASKRUN: C:\Program Files (x86), 0 found
2022-04-26T11:05:55:TASKRUN: C:\Users\Az-Administrator\AppData\Local, 0 found
2022-04-26T11:05:55:TASKRUN: Scanning the following locations for Shortcuts:
2022-04-26T11:05:55:TASKRUN: C:\Users\Az-Administrator\Desktop, 0 found
2022-04-26T11:05:55:TASKRUN: C:\Users\Public\Desktop, 0 found
2022-04-26T11:05:55:TASKRUN: C:\ProgramData\Microsoft\Windows\Start Menu, 0 found
2022-04-26T11:05:55:TASKRUN: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup, 0 found
2022-04-26T11:05:55:TASKRUN: C:\Users\Az-Administrator\AppData\Roaming\Microsoft\Windows\Start Menu, 0 found
2022-04-26T11:05:55:TASKRUN: Post-Install Scan complete
2022-04-26T11:05:55:TASKRUN: End Task [Install Remote Desktop] with Status [Failed]
2022-04-26T11:05:55:GATEWAY: FAIL Reason: Installation process exited with code 1603
2022-04-26T11:05:55:GATEWAY: Agent Task Ended: Status:[Failed], Name=[Install Remote Desktop]

```

On the AVD Suitability page, we get a helpful summary of the Azure Virtual Desktop compatibility testing results. Here we can see that the x654 package passed the 'as is' test. While it feels a bit 'Inception' to install the AVD client on an AVD host, the install clearly reported a success for single session environments...

Type	Application	AVD (as is)	Multi session	MSIX
MSI	Remote Desktop	Failed	Unsuitable	Unsuitable
MSI	Remote Desktop	Passed	Unsuitable	AVD Ready

However we can see that the package reported a failure for multi-session environments. Again, we can use the helpful web-based console to discover the reasons...

```

Console Video Output Performance
2022-04-26T11:28:24:GATEWAY: Agent Task Started: [Multi-Session Smoke Test msrdcw.exe]
2022-04-26T11:28:26:TASKRUN: Start Task [Multi-Session Smoke Test msrdcw.exe]
2022-04-26T11:28:26:TASKRUN: Starting Multi-Session Smoke Test for [2] user session(s)
2022-04-26T11:28:26:TASKRUN: Creating users for Multi-Session Smoke Test
2022-04-26T11:28:31:TASKRUN: Connecting to [AVDEnvironVM2] as User [RimUser2], Session ID [2]
2022-04-26T11:28:36:TASKRUN: Connecting to [AVDEnvironVM2] as User [RimUser3], Session ID [3]
2022-04-26T11:28:42:TASKRUN: Waiting for [2] Multi-Session Smoke Test(s) to complete
2022-04-26T11:29:11:TASKRUN: Starting Multi-Session Smoke Test for User Session ID [3], Target Application Path: [C:\Program Files\Remote Desktop\msrdcw.exe]
2022-04-26T11:29:11:TASKRUN: Starting Multi-Session Smoke Test for User Session ID [2], Target Application Path: [C:\Program Files\Remote Desktop\msrdcw.exe]
2022-04-26T11:29:11:TASKRUN: Smoke test is running as the currently logged in user: [RimUser3]
2022-04-26T11:29:11:TASKRUN: Smoke test is running as the currently logged in user: [RimUser2]
2022-04-26T11:29:11:TASKRUN: Running smoke test for "C:\Users\Az-Administrator\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Remote Desktop.lnk"
2022-04-26T11:29:11:TASKRUN: Running smoke test for "C:\Users\Az-Administrator\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Remote Desktop.lnk"
2022-04-26T11:29:12:TASKRUN: Smoke test will start in 5 seconds
2022-04-26T11:29:12:TASKRUN: Smoke test will start in 5 seconds
2022-04-26T11:29:17:TASKRUN: Running smoke test for [C:\Program Files\Remote Desktop\msrdcw.exe]
2022-04-26T11:29:17:TASKRUN: Running smoke test for [C:\Program Files\Remote Desktop\msrdcw.exe]
2022-04-26T11:29:29:TASKRUN: Sending smoke test response. This session (User Session ID [3]) will now close.
2022-04-26T11:29:34:TASKRUN: Sending smoke test response. This session (User Session ID [2]) will now close.
2022-04-26T11:29:39:TASKRUN: Multi-Session completed with [2] failure(s):
Session 2 : No application started.
Session 3 : Exited with error. An instance of the service is already running.
2022-04-26T11:29:39:TASKRUN: End Task [Multi-Session Smoke Test msrdcw.exe] with Status [Failed]
2022-04-26T11:29:39:GATEWAY: FAIL Reason: One or more multi-session smoke tests failed. Refer to the messages above for more information
2022-04-26T11:29:39:GATEWAY: Agent Task Ended: Status:[Failed], Name=[Multi-Session Smoke Test msrdcw.exe]

```

Lastly, the Rimo3 engine automatically converted our application to a .MSIX package, which can be attached dynamically to AVD machines, avoiding the need to install the MSI into a master image or via tools such as InTune. This is a very handy feature, as the creation process for .MSIX packages can be complex, and involves the creation and assignment of certificates to the packaged applications. As well as simply confirming packages work, Rimo3 offers a performance comparison. This is extremely useful as part of a migration exercise, because confirming that an application works does not necessarily mean that it works well.

By performing a reference install on the current OS platform, and then re-testing against the target OS platform, Rimo3 is able to validate that system performance is not compromised by the change, and can alert you via the console if anomalies are detected. The system performance during the install process is even captured for reference.



In the past, I have worked with organisations who employed an army of contractors to conduct app migration testing. I have no idea what the cost of this exercise was, but I doubt it was cheap! In Rimo3, we have a tool which can automate the whole process – and I do mean automate. Many tools offer automated processing, but require lengthy configuration in order to setup your routines. Rimo3's SaaS offering requires zero configuration to return the results presented above. The service has a set of predefined objectives, and all are useful and relevant to migration projects.

So if you or your clients are looking to better manage the application and OS lifecycle, migrate to Azure Virtual Desktop / Windows 365, or even just make a move from Windows 10 to Windows 11, then Rimo3's services could help to shorten timelines, improve visibility and reduce costs.