# Release notes (public preview)

## (public preview) RD Agent crashing

Certain conditions can cause the RD Agent to crash. When RD Agent crashes it does not gracefully handle mounting and staging MSIX images.

**Workaround:**

Option 1 (recommended): Restart session host VMs

Option 2: manually unmount all MSIX images either via Disk Management or script ([sample](https://github.com/stgeorgi/msixappattach/tree/master/unmount%20vhds))

## (public preview) Account with admin permissions destages MSIX packages

When the last user on a session host VM is admin and logs off or is being logged off due to inactivity this will cause not only deregistration of the MSIX packages but also their destaging. This will prevent following users from getting the applications registered.

Restage packages using PS

The **$path** variable is the path to the volume which contains the MSIX image and can be obtained via **mountvol** or **Disk Management.**

Replace **\\?\** with **file://**

Append **/apps/<packageName>/AppxManifest.xml** to thefile path**.**

[Windows.Management.Deployment.PackageManager,Windows.Management.Deployment,ContentType=WindowsRuntime] | Out-Null

Add-Type -AssemblyName System.Runtime.WindowsRuntime

$asTask = ([System.WindowsRuntimeSystemExtensions].GetMethods() | Where { $\_.ToString() -eq 'System.Threading.Tasks.Task`1[TResult] AsTask[TResult,TProgress](Windows.Foundation.IAsyncOperationWithProgress`2[TResult,TProgress])'})[0]

$asTaskAsyncOperation = $asTask.MakeGenericMethod([Windows.Management.Deployment.DeploymentResult], [Windows.Management.Deployment.DeploymentProgress])

$packageManager = [Windows.Management.Deployment.PackageManager]::new()

$path = “file://Volume{volumeguid}/apps/packageFullName”

$path = ([System.Uri]$path).AbsoluteUri

$asyncOperation = $packageManager.StagePackageAsync($path, $null, "StageInPlace")

$task = $asTaskAsyncOperation.Invoke($null, @($asyncOperation))

$task

## (public preview) How can I create CIMFS file to host MSIX image?

This feature is available in public preview.

## Changing what group, a user is assigned while user is logged in

When changing the user group membership while the user is logged in can lead to inconsistencies with the MSIX app attach applications that are to be registered and deregistered. It is recommended no change assignments while the user is logged in.

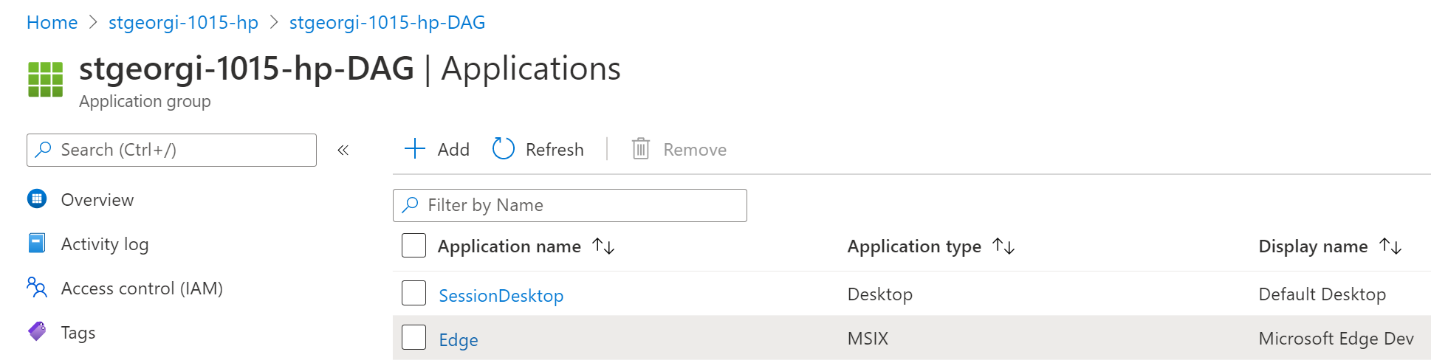
## MSIX package has multiple entry points (applications) and when I publish one all of them appear

This is the expected behavior. MSIX packaging format executes all operations (stage, register, deregister, and destage) per package and not per app (entry point).

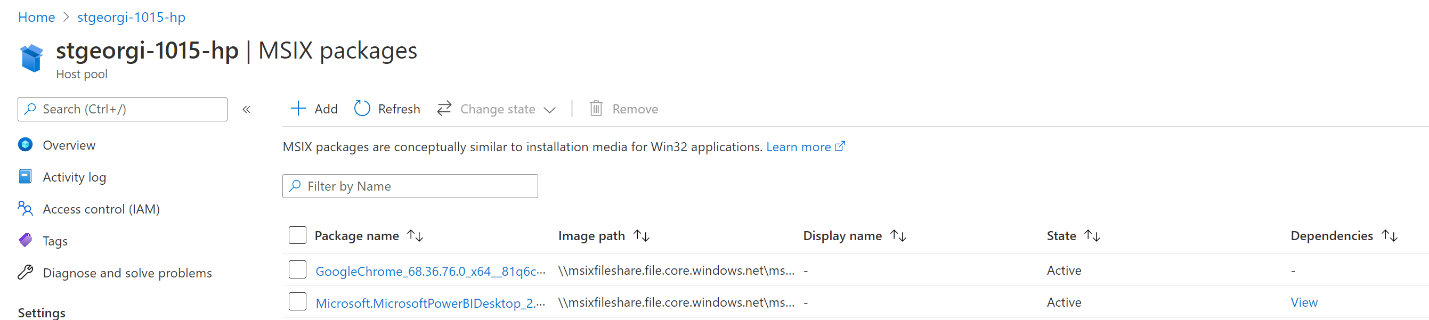
If there is need to break up the MSIX package use the MSIX packaging tool.

## Removing a package does not remove published applications

Microsoft Edge Dev is present as MSIX application.



But is missing in the MSIX packages grid.



This is by design. When a package is removed that might be triggered by the user looking to provide new version of the package. To avoid having to republish applications we do not automatically remove such “orphaned” apps.

## Same user has been assigned to both RAG and DAG

Scenario: a single user is assigned to a both remote application group (RAG) and a desktop application group (DAG) on the same host pool. Both groups have been configured to deliver MSIX applications via MSIX app attach. When that user accesses his/her DAG MSIX app attach will deliver MSIX applications from both groups not only the DAG.

This is by design.

## Application was removed for desktop application group, but user still sees it in their session

This is the expected behavior. If we terminate an application while it is being used this can lead to data loss. Application changes will only take effect after the user has logged off and logged back in.

## Application was added for desktop application group, but user does not see it in their session

This is the expected behavior. Application changes will only take effect after the user has logged off and logged back in.

## Publishing one application from an MSIX package

MSIX package can have multiple apps. When an MSIX package (with multiple apps) has been assigned to a DAG and that user access the DAG he/she will have access to all applications in that MSIX package.

This behavior is inherited from MSIX.

## End users need to log off to see newly published MSIX application when using RD

This is the expected behavior. Users are required to logoff and login to see new applications.

## Certain customer applications require elevation to start running services

This is the default behavior for MSIX packages. More info [here](https://docs.microsoft.com/en-us/windows/msix/packaging-tool/convert-an-installer-with-services).

## Dependencies

MSIX applications can have dependencies. More info on dependencies [here](https://docs.microsoft.com/en-us/windows/msix/desktop/managing-your-msix-deployment-overview#dependencies). All dependencies an MSIX application has must be addressed by the admin. There is no mechanism to auto resolved (deliver) dependencies.

While in some cases putting application dependencies in the same MSIX package as the application is possible this is not recommended. There is no guarantee that the dependency will be delivered prior to the application.

## Handling licenses

MSIX app attach is application delivery mechanism. It does not handle and enforce licensing.

## Dynamically staged MSIX packages do not show up right away

The RD Agent checks in with the RD Broker every 5 minutes to see if new packages have been added to the host pool. At the 5 minute mark, all dynamically staged MSIX packages will appear in Disk Management/Mountvol.

Same applies for when an MSIX package has been removed from the host pool. It will take up to 5 minutes to be removed from Disk Management/Mountvol. To prevent users from using the MSIX package it must be set to “Inactive” in the management UI.

## Manual restart of RDAgentBootLoader

When the RDAgentBootLoader service is being restarted it will cause destaging (on stop) and staging (on start). MSIX app attach is capable of gracefully handling the restart with version 1270 of the RD Agent.

## MSIX app attach remote open opens explorer window instead of the app

The RD Agent checks in with the RD Broker every 5 minutes to see if new packages have been added to the host pool. At the 5 minute, mark all dynamically staged MSIX packages will appear in Disk Management/Mountvol. If the IT pro publishes a remote app before those 5 minutes interval and user click on the app Explorer will try to start an application that is not available on the machine and display itself as the MSIX application cannot be found.

If more than 5 minutes have passed the application is behaving as described verify registration, dependencies, certificate, and staging status.

## User with reader permissions can see the MSIX app attach UI

This is inherited from the WVD service and Azure RBAC.