AppVentiX 5.1 Admin Guide

Table of contents

AppVentiX Quick Start 2

Deploy, Update and Remove applications with AppVentiX 9

AppVentiX Components 11

- **AppVentiX Agent Service** 11
- **AppVentiX Agent GUI** 13
- **AppVentiX Agent Refresh Shortcut** 14
- **AppVentiX Central View Console** 15
  - **Machine Groups** 15
  - **Machine Group Agent Settings** 16
  - **Manage Machines page** 24
  - **Manage Content page** 25
  - **MSIX and MSIX app attach** 30
  - **MSIX Shared containers** 32
  - **MSIX Certificate management and deployment** 32
  - **Application overview** 33
  - **FSlogix App Masking management** 34
  - **App Control setup and best practices** 35
  - **Configuring User Settings** 40
  - **Configuration and Activity page** 44
  - **Firewall \ communication ports used by AppVentiX** 45
  - **Limit access to the Central View console** 45
  - **Audit trail** 46
  - **Azure Virtual Desktop (AVD) integration** 47
  - **Azure AD (Entra ID) integration** 48
  - **Azure file share configuration** 51
  - **How to setup a QUIC share to allow share access over port 443** 57
  - Firewall Rules 57
  - User Accounts 58
  - Create File Structure 58
  - NTFS Permissions 59

SSL Cert 60
Advanced configurations 62
Share permissions and share configuration 62
Central View inventory 64
Central View advanced settings 64
Supported operating systems 65
Upgrade from earlier version of AppVentiX 66
FSLogix and roaming profile settings 66
Automated image building and deployment actions 68
Image build events 68
Run the refresh cycle from the command line 69
When Central View console takes longer to start 69
Example configurations of the AppVentiX Agent 70

AppVentiX Quick Start

AppVentiX works with a lightweight agent running on the same machine as the App-V and MSIX (app attach) client (part of Windows). This can either be a virtual machine (Microsoft RDS\AVD Hostpool\Windows365, Citrix VDI (PVS\MCS), VMware Horizon, etc) or physical machine (PC or Laptop). The agent can be pushed from the Central View console or easily installed silently, the silent install parameter is already populated in the Central View console and you only have to copy\paste the command from there.

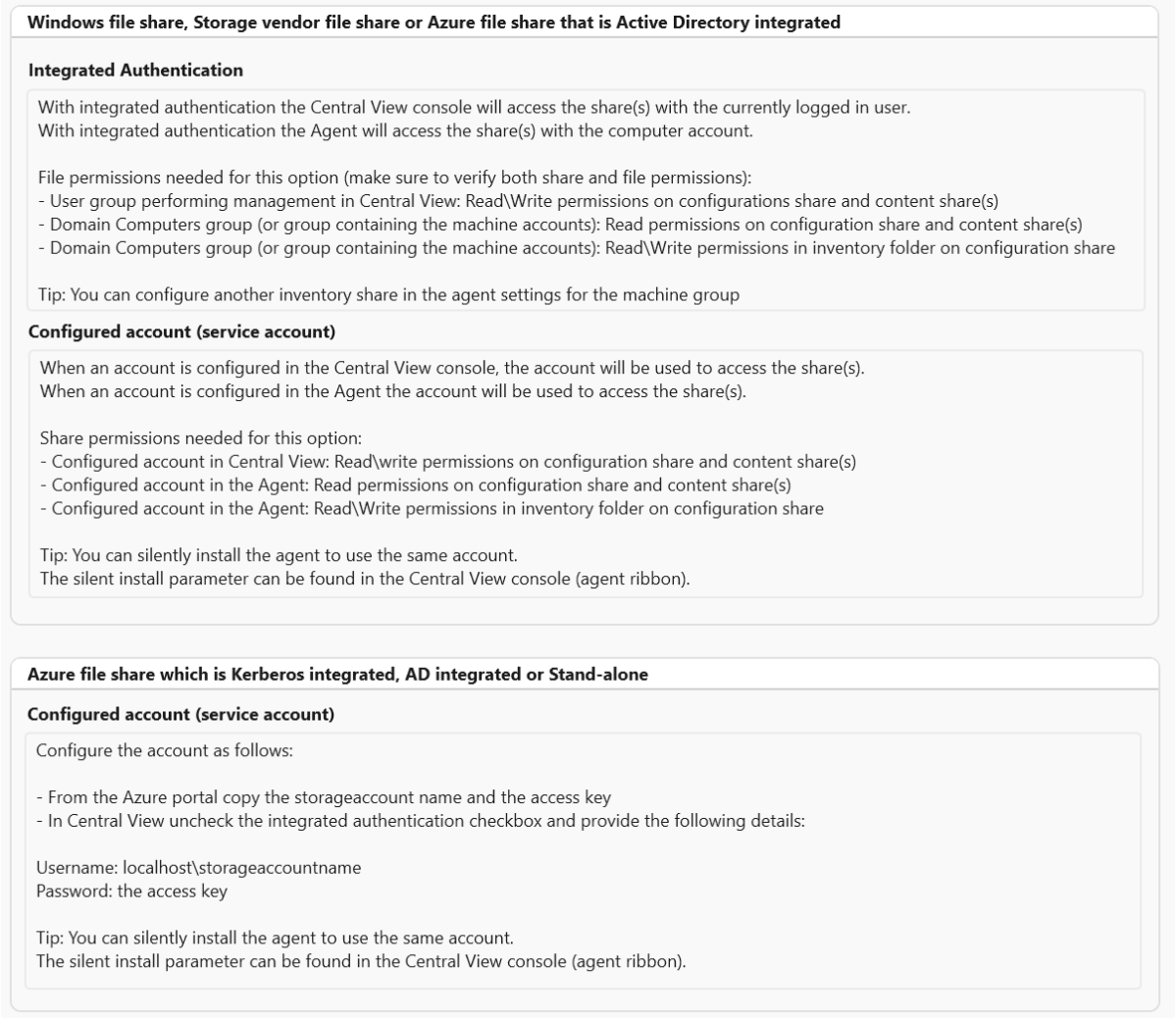
The Central View console can be installed on any management machine and does not need any (SQL) back-end. AppVentiX only requires a file share, a lot of file share types are supported and proven to work with AppVentiX, to name some examples: Windows file shares (direct or DFS), Azure file shares (domain integrated and stand-alone), Nutanix and NetApp file shares, and QUIC shares accessed over port 443. The Central View console requires a 64Bit OS (Server OS or Desktop OS). Both Server OS, Multi-session OS and Single-session OS are supported by AppVentiX. The AppVentiX solution is easy to implement and will give you complete control and insight in under 10 minutes.

The quick start steps will begin on the next page.

The following steps will help you to get up and running quickly:

Step 1

Create a new file share or use an existing one, this share will be used to store the central configuration. You can use a normal Windows file share or a DFS share to make the share high available (for example \\yourdomain.local\appventix\config). Also Azure file shares (domain integrated or stand-alone) and shares on storage vendors like Nutanix\NetApp\Dell\HP file shares are supported. AppVentiX also supports QUIC shares, which are shares that can be accessed over port 443 using highly encrypted TLS. You can read more about setting up a QUIC share or Azure file share in this admin guide. The required share permissions can be found in the Central View console in the settings window, a screenshot is added below. Basically you can choose to use **integrated authentication** or provide a **(service) account**.



For integrated authentication (default), you can configure the below permissions:

- Domain computers group read permissions on the share
- Domain computers group read\write permissions on inventory folder on the share (this folder is automatically created)
- Central View Admins (a group with users that performs the management) read\write permissions

Step 2

Create a content share (for example \\yourdomain.local\appventix\content), this share will be used to store the packages on. You can also create a folder on the same share as the configuration share created in step 1. The share permissions are the same as in step 1.

For integrated authentication (default), you can configure the below permissions for the content share:

- Domain computers group read permissions on the share
- Central View Admins (a group with users that performs the management) read\write permissions

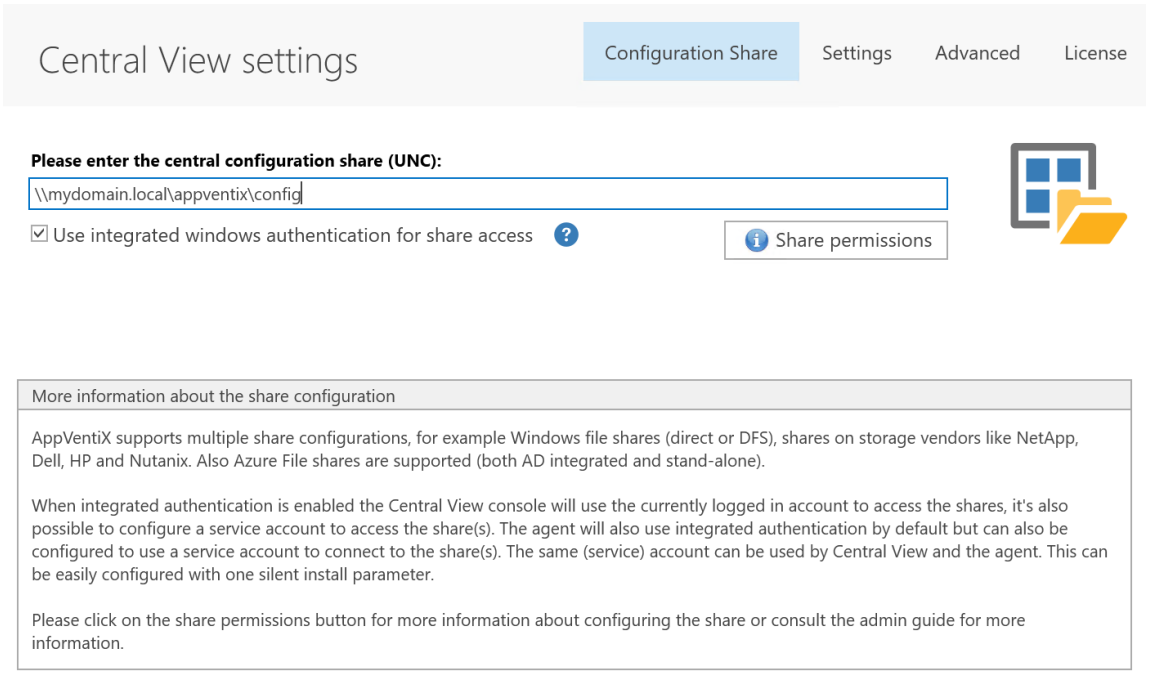
When using a (service) account you can use the permissions from the screenshot in step 1.

Step 3

Install the AppVentiX Central View console on any machine you like, only keep the following in mind:

- The machine has to be installed with a 64 bit OS (Server OS or Client OS)
- The configuration and content share needs to be accessible

After you have installed the Central View console, click on the icon to start the console. The first time configuration window will open:



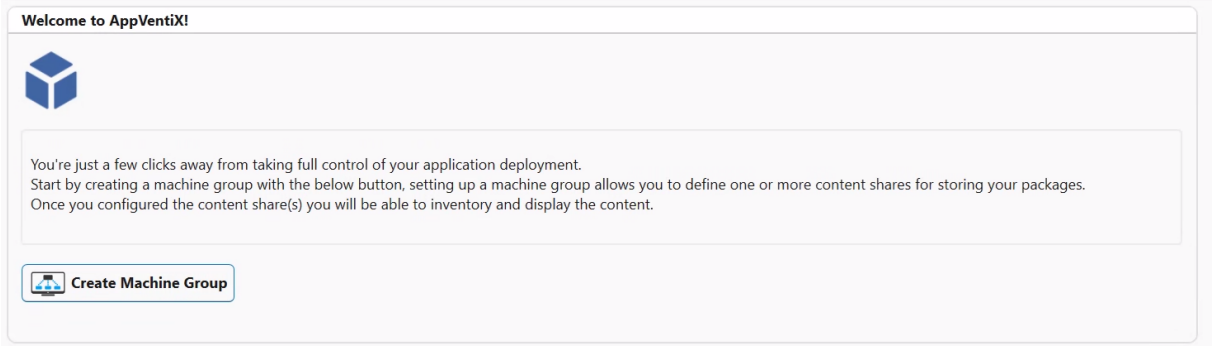
Configure the share you created in Step1. You can enter any UNC path you like.

Note:
Most of the time the default Central View settings are sufficient, optionally you can configure additional settings when needed, you will find more information about the settings in this guide.

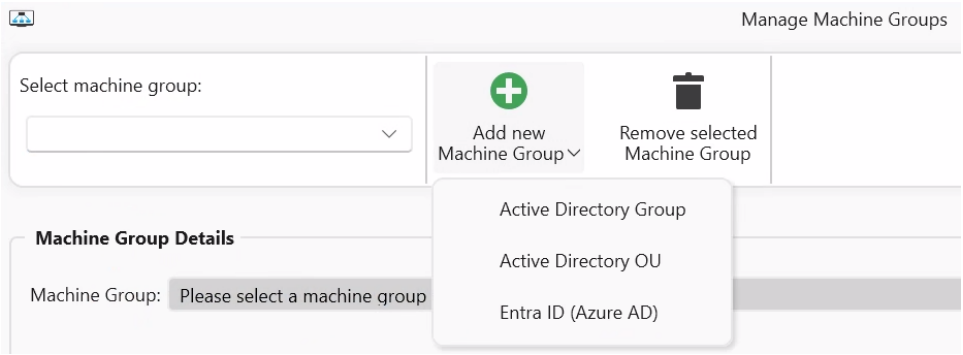
Click save.

Step 4

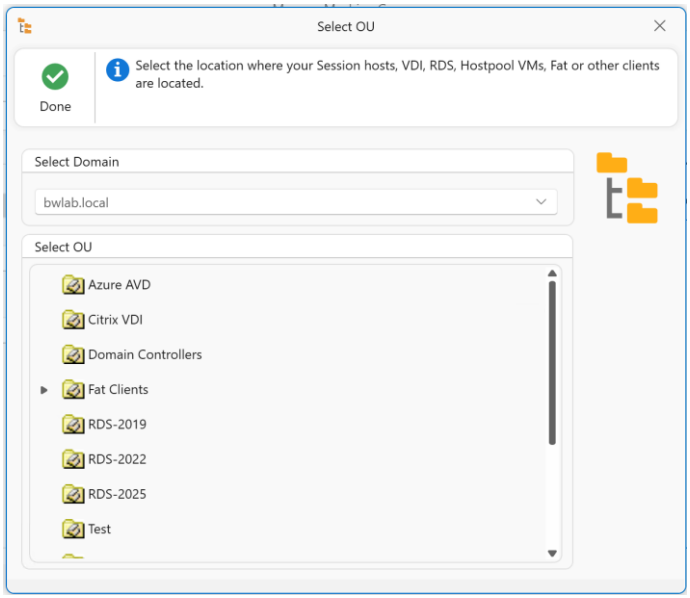
Now we will create a Machine Group, you can use machine groups to implement DTAP (test, acceptation, production environment) or separate your deployment for hybrid use cases (laptops\ virtual desktops, etc). A machine group is just a reference to an AD OU, AD Group or Azure AD tenant, there is no need to update this groups manually.



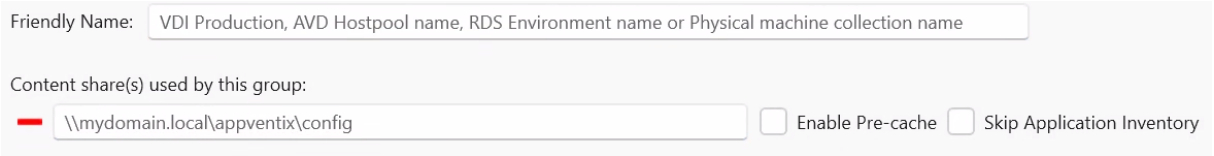
Select the option where the machines which you want to manage are located.



In this example we will create a machine group based on OU:

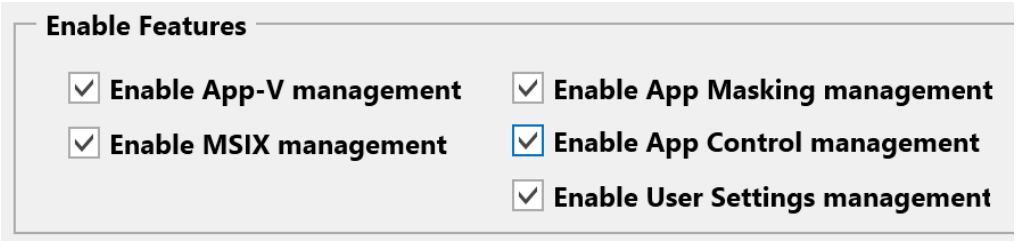


Select an OU and click Ok.



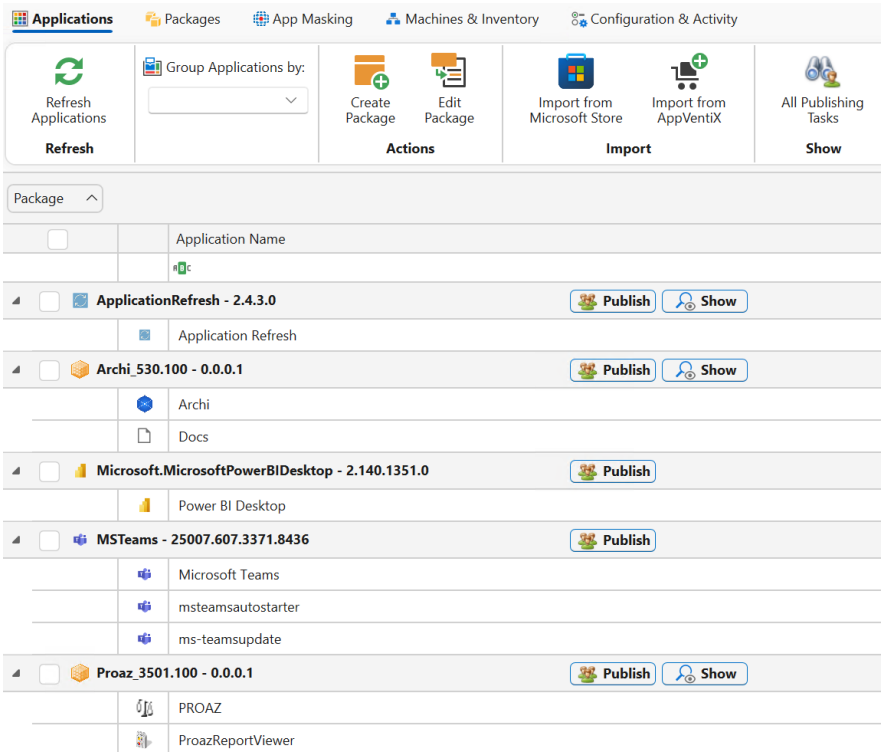
Provide an easy to remember friendly name like RDS Production or name it the same as the Citrix Delivery group, AVD Session Host pool or group of physical machines. The content share is automatically pre-populated with the same share as the configuration share, the content share is where the packages\containers are stored. Check if the content share is accessible and configure multiple content shares if you wish.
When the pre-cache checkbox is enabled, the agent will preload (App-V) \ prestage (MSIX) packages in the cache when the machine boots or when the refresh cycle is triggered. If you disable pre-cache they will be added on the fly whenever a user needs the package, making it a real dynamic delivery mechanism. You can use a combination of both approaches: pre-cache certain packages and dynamically delivery others, you can do this by creating 2 content shares, one with pre-cache enabled and another one with pre-cache disabled. The skip application inventory checkbox will hide applications from this content share in the Central View applications overview window.

After configuring the content share(s) click on Configure Agent Settings.



Enable the feature(s) you want to use and click Done. In this quick start guide we will use the default settings for each feature. Please check the Agent Settings chapter for an explanation of all agent settings. **Most of the time the default agent settings are a good starting point. For MSIX it's only important to choose the correct profile setting, for profile containers like FSlogix keep the default setting, if your machine group is for a fat client \ laptop environment, select local profile.**
Click on Save Machine Group and close the Manage Machine Groups window.

The first time inventory of the content share is now executed, if no content is found you are asked if you want to import some packages to get started.
You can now click through the console and check if you can see the content and if the machines are visible when you select the machine group in the manage machines window:

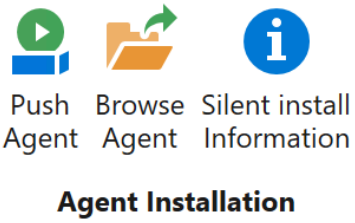


Step 5

Now that the Central View console is up and running we will install the agent. The agent can be pushed remotely to the machines (this can even be done when users are logged in, no reboot is needed) or you can install the agent silently using an automated procedure, like an image build procedure or pipeline. When an older version of the agent is detected, it will be upgraded automatically.

Push agent:

Select one or multiple machines and click on the Push Agent button (manage machines page):



The agent will now be installed or upgraded automatically.

Manual install:

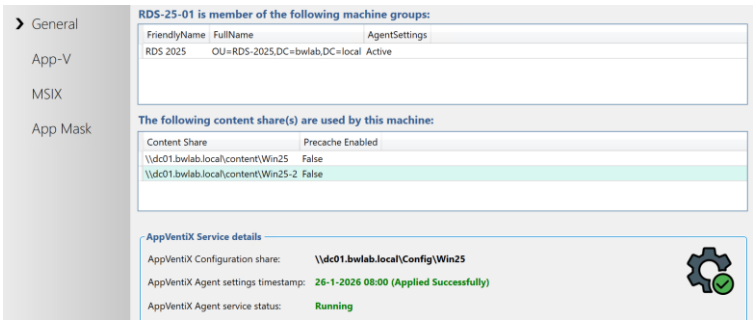
Click on the Browse Agent button, the agent installation is now shown.
Double click the installation on a machine and follow the installation prompts.

Silent install:

Click on the silent install information button, the silent install parameter will be shown.

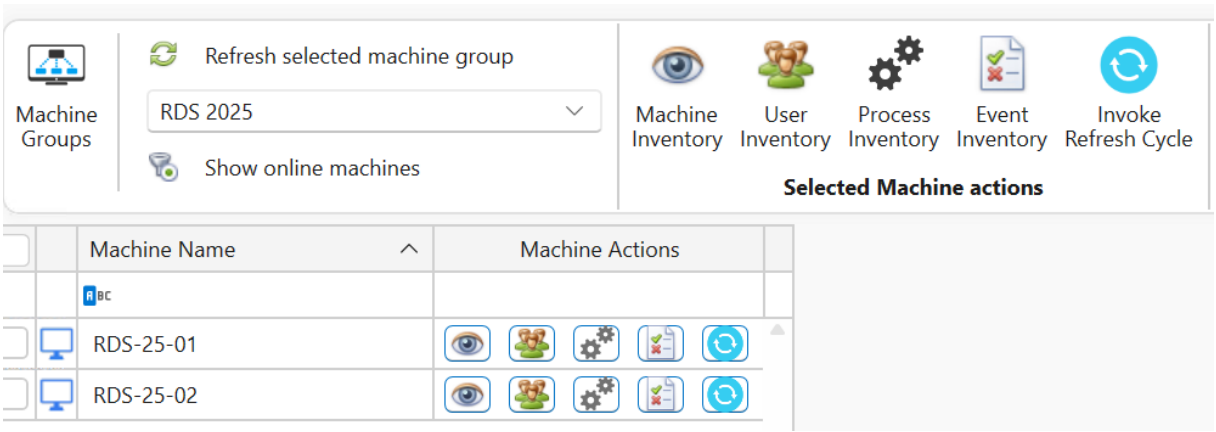
The agent contains a small GUI (the AppVentiX Agent GUI), a shortcut to this GUI is not placed in the startmenu, you can find it in: “C:\Program Files\AppVentiX\AppVentiX Agent”. The GUI will show you the service state and detected machine group, if the machine group is not detected or another error is displayed, please open the agent log (button at the right) it will show you a lot of useful information.

The AppVentiX Agent GUI is great for checking the service state, but also to see which packages are deployed, see package details and to troubleshoot and manage them. You can also invoke a refresh from the agent GUI (button at the bottom). A refresh can also be invoked in the Central View console (per machine or machine group). The refresh cycle runs automatically when the machine starts, optionally it can also be configured with a timer in the agent settings. By default user publishing is refreshed when the user logs in and user publishing will also be refreshed when the refresh cycle is invoked. This settings can be adjusted in the agent settings for the machine group.

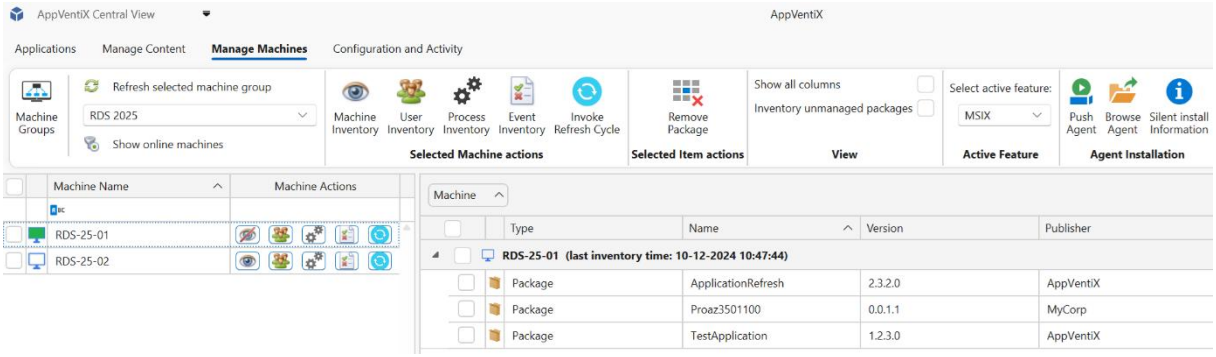


Step 6

You’re now ready to deploy packages, shortcuts, user settings, App Masking rules, and/or App Control policies, and manage your deployment.
Return to the Central View console to see how easy it is to inventory and manage machines by clicking the eye icon. During these quick steps, you configured a content share for the machine group. If the Pre-cache checkbox is enabled, packages from the content share will be preloaded onto the machine(s) when you click the Refresh Cycle button. If it’s disabled (default), packages will be delivered based on the publishing tasks you configure. The Refresh Cycle (blue circle) updates publishing for currently logged-in users, pre-caches new packages, and automatically removes packages that are no longer in the content share, no manual cleanup required.



By inventorying the machine using the eye icon, you can view the App-V or MSIX packages, App Masking rules, and App Control policies currently loaded on the machine:



In the machine inventory view you can right click on the column header and select filtering options to easily find a package. With the user inventory feature (user group icon) you can see in real-time which users are logged in and which packages they have published.

With the “Show online machines” button you can filter the machines that are online (they will become green) and you can also see the agent version as well.

Next steps

- Go to the manage content page in Central View, check the content on your configured content share and create publishing tasks (assign packages to users or globally to machines). A great feature of AppVentiX is that unmanaged packages (when you remove a publishing task or remove a user from AD group assigned to a publishing task) will be unpublished\removed automatically for users.
- Explore all Agent Settings, check the settings to meet your deployment goals, for example AppVentiX has different options to accommodate each scenario (for example persistent\non-persistent). Read through this admin guide to read more about configuration options, use cases and managing the deployment.
- Explore the Central View console and Agent GUI and check out all the options and features.

Deploy, Update and Remove applications with AppVentiX

With AppVentiX it is easy to deploy and update applications in real-time. It is possible to run multiple versions of an application side by side or replace an application with a new version immediately. There are a couple of approaches, and they are largely the same for App-V and MSIX, we would suggest trying out the different approaches to get familiar with them, you will see results in real-time. Managing applications with AppVentiX puts you back in control and will make you confident about application deployment and updates.

Deploy a new application:

- Place the new package containing the application on the content share
- Create a publishing task for the application

- When a user logs on or when the user clicks on the Refresh shortcut in the start menu the new application is published automatically, the user will receive the new application without having to log off and on again.

Update an existing application, approach 1 (run old and new side by side):

- You can either choose to create a new package or update the existing package, save the package as new package after updating (to allow old and new version to run side by side)
- Place the package containing the updated application on the content share
- Create a publishing task for the updated application (you can first assign this updated application to a test group for example)
- When a user logs on it will receive the new updated application, also when you run the refresh cycle (can be invoked centrally or on the machine itself) it will receive the updated application without having to log off and on again
- After your test users have verified the updated application, you can edit the new publishing task and assign the production group
- Remove the old publishing task, after doing this the old version will be removed for the user automatically

Update an existing application, approach 2 (run new version on test machine group):

- Copy the updated application to a content share which is configured for your test machine group
- Create a publishing task and filter this to only apply for the test machine group
- Let users log in to the test machine group so they can test the application
- Copy the package to the production content share after tests has finished
- Edit the publishing task so it will also apply on the production machine group

Update an existing application, approach 3 (replace old version with new version):

- Save the package as new package after updating or save the package with the same package name\id to increment the version number
- Place the package containing the updated application on the content share
- Create a new publishing task for the updated application and assign the same group as the old publishing task from the previous version
- Already logged in users will receive the updated application when the refresh cycle runs and new users when logging on, the already active users can continue to work in the old version. The new version will be active automatically when the user closes and re-opens the application
- If you want to immediately replace the old version with the new one (force), remove the old publishing task (or configure the force upgrade option in the new publishing task), the old version will be closed and the new version is active immediately. Please note that the application will be forcible closed, so when the user should keep the old version open leave the old publishing task in place and remove it later

Remove application, approach 1 (remove application for certain users):

- Remove users from the AD group assigned to the publishing task
- The application will be automatically removed for the users that are removed from the AD group

Remove application, approach 2 (remove application for all users, soft remove):

- Remove the publishing task for the package
- The application will be removed for the users
- After a while remove the package from the content share, it will be removed from the cache on the machines by the balance cache mechanism of AppVentiX

Remove application, approach 3 (remove application for all users, hard remove):

- Remove the publishing task for the package
- The application will no longer be published for users
- Remove the package from the content share and invoke the refresh cycle, this will prevent the package from being pre-cached and it will be removed from the cache on the machines by the balance cache mechanism of AppVentiX

Remove application, approach 4 (remove application using the drain mechanism):

- Remove the publishing task for the package
- The application will no longer be published for users
- Open the package options for the package which you want to remove, select drain this package. The package will no longer be deployed and removed from the cache when the refresh cycle runs. You can leave the package on the content share.

Remove application, approach 5 (by using inventory):

- Inventory the machine(s)
- Filter the package you want to remove on name
- Select the package (on one or multiple machines)
- Click remove package, the package will now be removed immediately

AppVentiX Components

AppVentiX consists of 3 main components:

- The AppVentiX Agent (Service, Agent GUI and Refresh shortcut)
- The AppVentiX Central View Console (Central management)

Please find more information about installing the agent in the quick start steps.

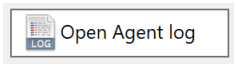
AppVentiX Agent Service

The service is responsible for deploying and managing packages, for publishing packages for users that are logging on and for users that are already logged on. The agent makes smart decisions, it will only publish new packages to users and it will unpublish packages for users automatically when they are no longer managed (when you remove the publishing task or remove a user from a group for example). The agent service can be configured with Agent Settings to fine tune your deployment. This settings will be discussed in detail later in this guide.

Every action of the service is logged in a dedicated eventlog, to make troubleshooting easy and will give you good insight in your AppVentiX deployment.
The AppVentiX Agent eventlog can be found directly under Applications and Services Logs :



You can open the eventlog very easily from the Agent GUI with the Open Agent log.



You can also inventory the log remotely from Central View console, by clicking on the log icon next to a machine:

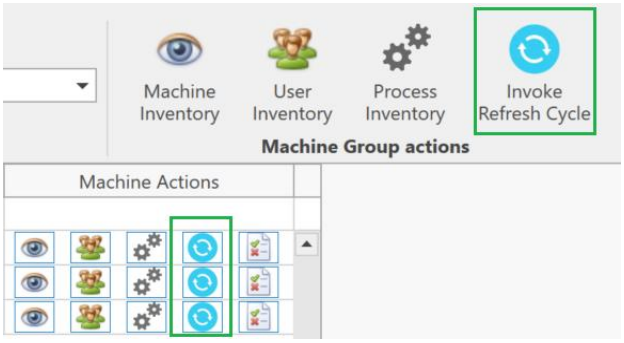


The agent service is event driven and the actions can be configured with Agent Settings. For example you can configure the agent to clear the cache at machine start, publish packages at user login etc. One of the most important events is the refresh cycle:

The Refresh cycle
The refresh cycle handles the deployment of new packages by comparing which ones are already present in the cache with the ones on the content share(s), new packages will be cached when the pre-cache option is enabled for the content share. The refresh cycle runs at machine start-up and while the machine is running it can be triggered in multiple ways:

- Through a configurable timer (in the Agent Settings)
- Manually on the machine through the AppVentiX Agent GUI (Run Refresh cycle button)
- With the following Powershell command: (Get-Service 'AppVentiXService').ExecuteCommand(252)
- Remotely by the AppVentiX Central View console
- The user can click on the Refresh Workspace shortcut in the start menu

You can recognize the Refresh Cycle by the blue circle:



In Central View you can invoke the refresh cycle for a single machine (blue circle next to the machine) or for the whole machine group (machine group actions).

The refresh cycle does the following things on the agent:

1: Pre-cache packages from content shares (only when pre-cache is enabled for the content share) (App-V and MSIX)
2: Remove packages from the cache that are no longer on one of the configured content shares (when this setting is enabled in the agent settings). Default this setting is enabled. (App-V and MSIX)
3: Execute global publishing tasks (App-V only)
4: Execute user publishing tasks for currently logged in users (App-V and MSIX), deploy app mask rules, app control policies and refresh user settings

Optionally the below PowerShell commands can be used to initiate the refresh cycle, for example as scheduled task or an automation solution:

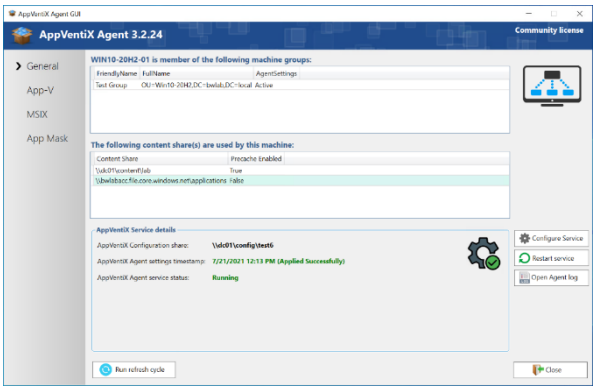
Refresh cycle action	**Powershell Command**
Refresh both cache and user publishing tasks (default)	(Get-Service 'AppVentiXService').ExecuteCommand(252)
Refresh only cache: balance cache to remove old packages, pre-cache packages when enabled for a content share, process package drain tasks	(Get-Service 'AppVentiXService').ExecuteCommand(242)
Refresh only user publishing tasks for currently logged in users and refresh global publishing tasks (App-V)	(Get-Service 'AppVentiXService').ExecuteCommand(241)

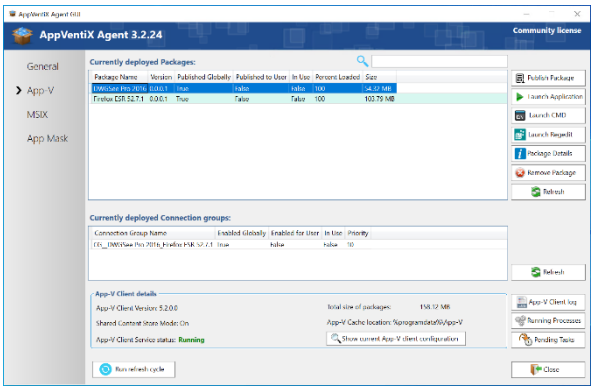
AppVentiX Agent GUI

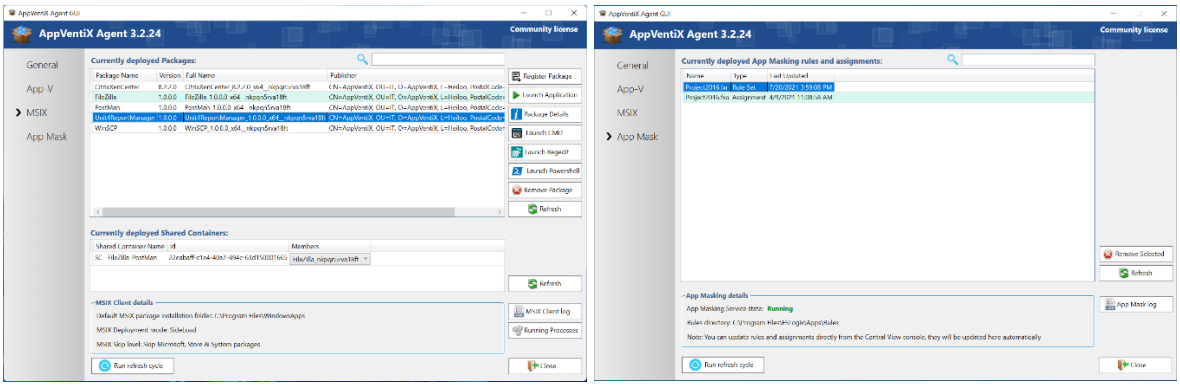
The AppVentiX Agent GUI can be used for:

- Checking the service state
- See details about deployed packages, app mask rules & app control policies
- Troubleshooting
- Invoke a refresh

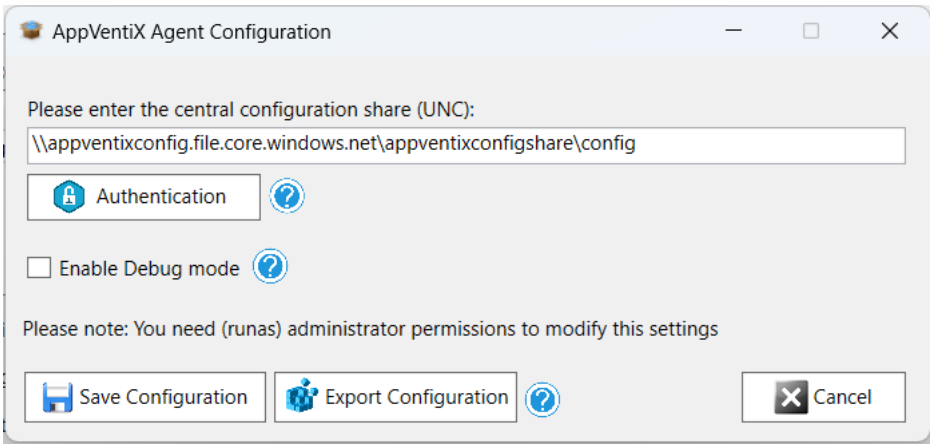
Example screenshots of the Agent GUI:







The AppVentiX agent is configured centrally in the Central View console. There are a couple of settings you can configure when you click on the Configure Service button:



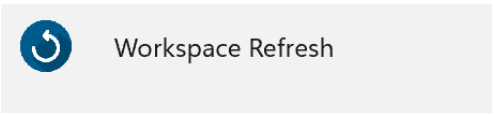
In the agent configuration you can change\update the configuration share. You can also provide a user account which the service will use to access the configuration share and content share(s). By default the service will use integrated authentication.

Debug mode will log more information to the eventlog for troubleshooting purposes, make sure to disable this option after troubleshooting.

You can export these settings to a registry file to import on other machines. This settings can also be provided for silent installation of the agent, click on the silent install button in the Central View console (agent ribbon) to see the prepopulated silent installation parameter. Please note that all other Agent Settings are configured and stored centrally.

AppVentiX Agent Refresh Shortcut

The following shortcut is placed in the start menu, the user can click on this refresh shortcut to execute a refresh:



A refresh progress will be displayed:



This refresh updates applications, App Masking rules, App Control policies, and user settings.

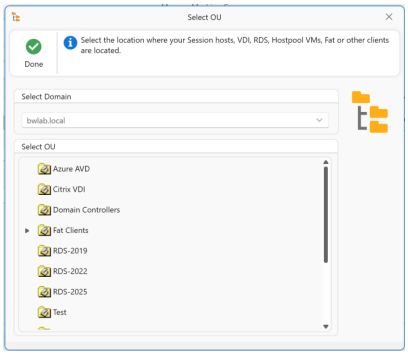
AppVentiX Central View Console

Central View is the centre piece of your deployment. It is a lightweight, easy to use real-time management console. Central View does not need a dedicated back-end server.

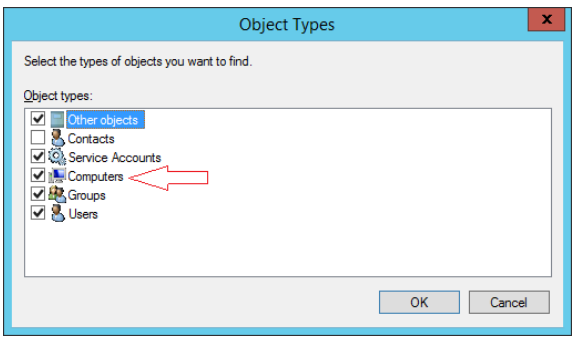
Machine Groups

Central View reads Active Directory group(s), Active Directory OU's or Entra ID to retrieve the machines in a specific machine group. The agent will automatically detect in which machine group it belongs. Please read the quick start steps at the beginning of this guide to get an impression how to create a machine group.

In Active Directory environments, the easiest approach is to create machine groups based on OUs. Machines are automatically pulled from the selected OU, so there's no need to manually add or remove devices, the group updates automatically.



When creating a machine group based on AD group instead of OU, make sure you enable below option in Active Directory when adding machines to the AD group, or else you can't find machine accounts to add to the AD group:



Note:
You can create multiple machine groups and machines can be member of multiple machine groups, you can also configure multiple content shares for a machine group. The agent will retrieve packages from all configured content shares. The agent will only apply Agent Settings from the first Machine Group it is member of.

For Entra ID based machine group, please consult the Entra ID (Azure AD) setup section in this admin guide.

You can always edit a machine group and change Agent Settings, please note that the Agent Service on the machine needs to be restarted before the new settings will be activated.

When creating a machine group you can configure content share(s) for the group, next to the content share you have the option to select a checkbox: Enable - Precache.

When pre-cache is enabled for a content share the agent will deploy packages from the content share in the cache when the refresh cycle runs. Use pre-cache when you want packages to be available on the agent before a user logs in (for fast publishing). For App-V this options means packages are loaded on the machine, for MSIX this means packages are staged on the machine (when used in combination with app attach, the disk is attached and the package staged).

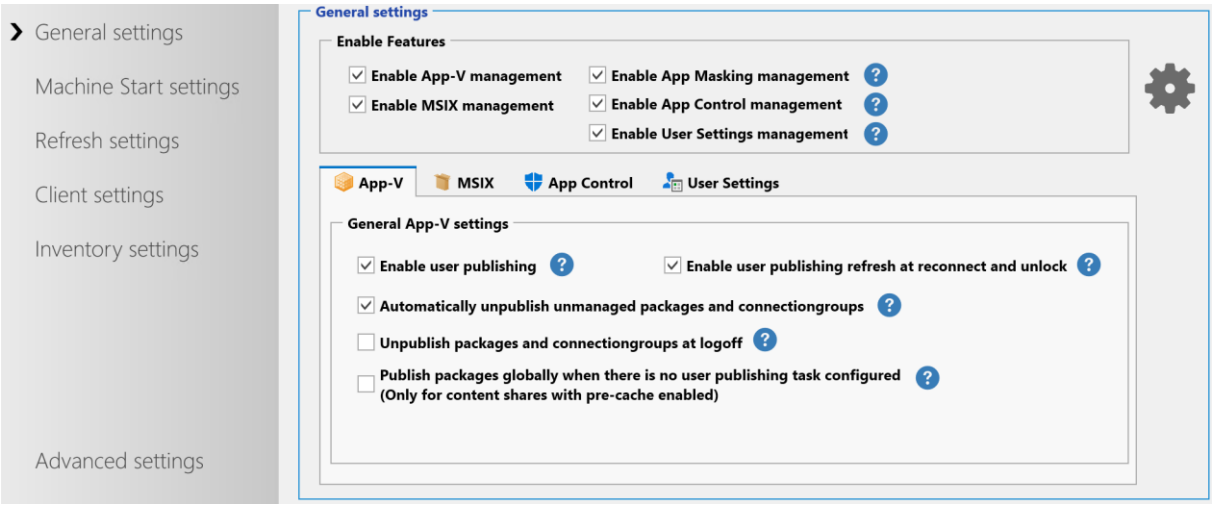
Don't enable pre-cache for a content share if you want packages to be deployed on the fly at user login using publishing tasks. This is supported for all package formats, App-V, MSIX + app attach.

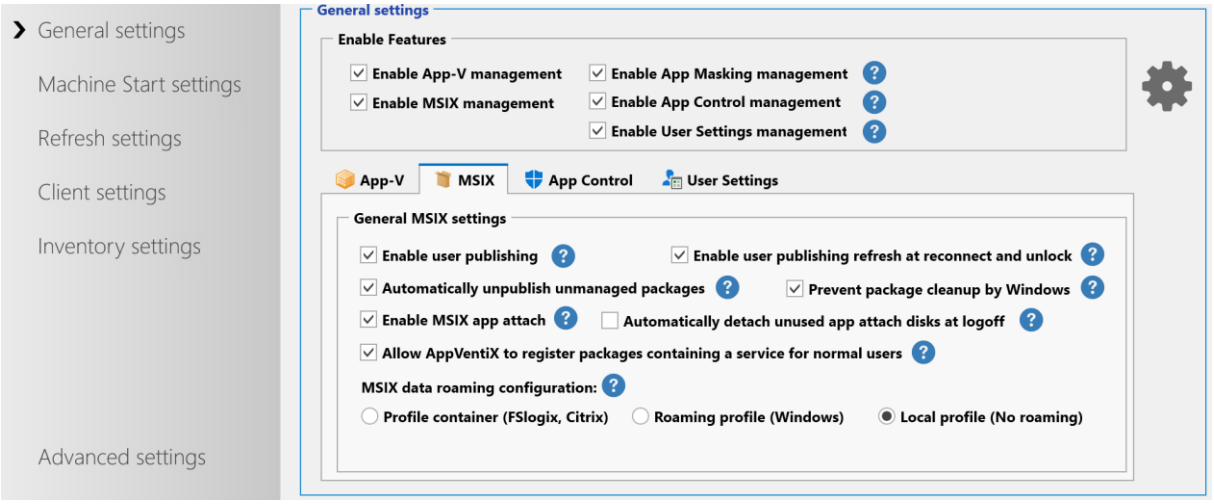
Machine Group Agent Settings

Configure Agent Options to fine tune your deployment. Agent options are retrieved and applied by the agent running on the client machine. Agent settings are applied when the agent service (re)starts.

General Settings

Here you can enable App-V and\or MSIX, you can also enable them both to use them side by side. The settings are split between the features to easily navigate through the settings.

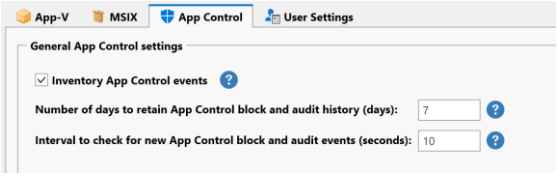




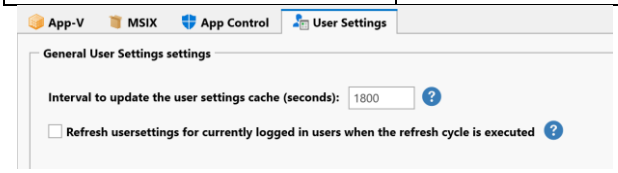
The most Agent Settings speaks for themselves, the tooltip will provide additional information by holding your cursor on the question mark icon. All settings not in each own tab applies to all features, where they are different you will find them in their own tab\groupbox.

General settings	
Enable user publishing	Enables the service to process user publishing tasks when a user logs in (App-V and MSIX)
Enable user publishing at session reconnect	Enabled the service to process user publishing tasks when a user reconnects or unlocks his session (App-V and MSIX)
Publish packages globally when there is no user publishing task configured	Instructs the agent to deploy packages globally when there is no user task configured for the package (App-V only)
Unpublish packages and connectiongroups at logoff	This setting is designed for environments where users frequently log in to different machines. It ensures packages are unpublished at logoff, preventing orphaned shortcuts or registrations if user assignments change or packages are removed while the user is logged off, maintaining a clean publish state.
Automatically unpublish unmanaged packages	This setting makes the deployment fully managed. When you remove a publishing task the package will be unpublished \ removed for the user automatically (App-V and MSIX)
Enable MSIX app attach	This enabled the app attach integration in the agent (MSIX only)
Automatically detach unused app attach disks at logoff	Default this setting is disabled. By default app attach disks are automatically detached when the machine is rebooted, optionally you can enable this setting if you want to detach unused app attach disks at logoff. When enabled, the service will check 30 seconds after a user logoff if the app attach disk is still in use, if not it will be detached. This allows you to cleanup old disks from the content share without rebooting the machine also the number of attached disks on a machine will be lower. This feature is especially useful for shared OS (like RDS \ multi-user OS) and when you have a lot of app attach disks to manage and the machines are rebooted less frequently. Events about the auto detach feature are logged in the AppVentiX eventlog.
Allow AppVentiX to register packages containing a service	MSIX packages containing a service can only be added with elevated permissions, AppVentiX can add the package to allow the

for normal users	service to install and will then register it for a normal user account
Prevent package cleanup by windows	When you use FSlogix or another profile container solution and a user logs off the profile is removed from the machine. Windows can detect that a package is no longer needed on the machine and removes the package. It will be deployed automatically again when the user logs in again, but with this setting the package will stay on the machine when the user logs off. Then you can remove packages using the balance cache feature of AppVentiX, which will automatically remove packages from the cache that are no longer on one of the content share(s).
MSIX data roaming	This setting will allow you to configure the profile solution you are using. By default the profile container option is enabled, this will make sure the package deployment is compatible with container based profile solutions. The retain user settings stored in the systemappdata folder makes sure these settings persist as well. Please contact support if you need more information about this setting.



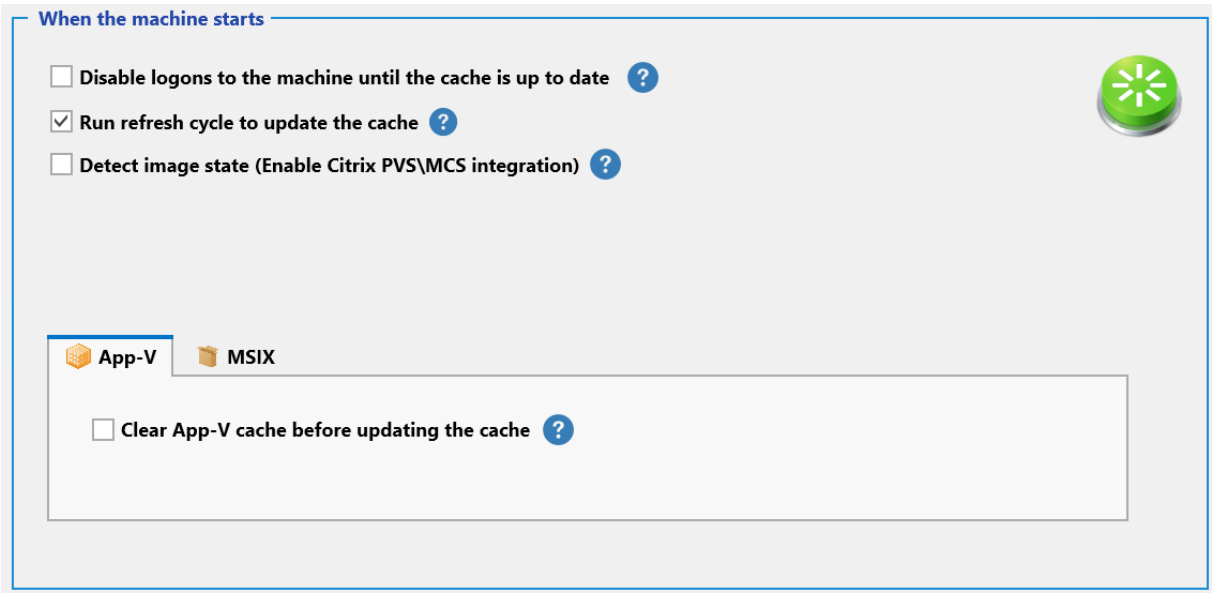
Inventory App Control events	This setting enables inventory and logging for App Control block and audit events. These events are stored centrally in the Inventory folder of the configuration share. In Central View, click the App Control Log button to view the logs. From there, you can create App Control policies directly.
Number of days to retain App Control block and audit history	Specifies the number of days that block and audit events are collected and stored before being overwritten by newer events
Interval to check for new App Control block and audit events	Specifies the interval for checking new audit and block events. Only events newer than the previous inventory are processed, ensuring efficient, low-impact checks. During learning mode, configure a shorter interval to collect events more quickly.



Interval to update the user settings cache	The time interval when the user settings cache is updated, when an assigned user setting is not found in the cache it will be retrieved online automatically. Optionally you can redirect the cache location to another drive in the advanced settings.
Refresh usersettings for currently logged in users when the refresh cycle is executed	By default user settings are only executed during user login or when the user clicks on the Refresh Workspace shortcut, when this setting is enabled, usersettings are also refreshed for logged in users when the refresh cycle runs on the machine.

Machine start actions

With machine start actions you can configure the machine start configuration.

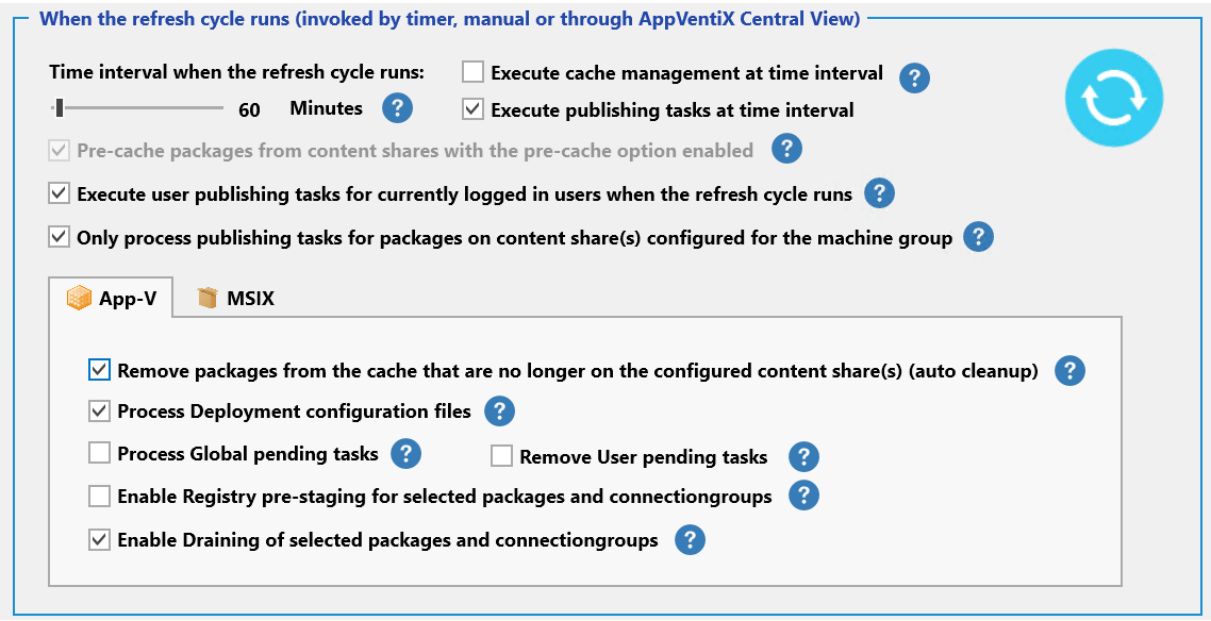


When the machine starts	
Disable logons to the machine until the cache is up to date	The agent will disable logins while the cache is updated at machine start. It will enable logins again when the cache is up to date
Run refresh cycle to update the cache	Invoke the refresh cycle at machine start to pre-cache packages from the content share(s) (when enabled for the content share)
Detect image state (Citrix PVS\MCS integration)	When the machine starts, the agent can detect the image state (read only or read\write). It can make smart decision how to deploy packages. Other agent settings are updated automatically when you enable this feature. There are 2 supported scenario's: **Scenario1:** Private mode: Stop Refresh cycle, this will prevent packages from being deployed while the image is in private mode. Read-only mode: Deploy packages This scenario is often used for non-persistent RDS environments, you can redirect the cache to a persistent drive. **Scenario2:** Private mode: Deploy packages in the cache Read-only mode: Deploy new packages using optimized delivery (App-V Shared Content Store mode or MSIX AppAttach) This scenario is often used for non-persistent VDI environments. Leave the cache in the default location. When the image is build or opened for Windows Updates for example, the agent will automatically pre-cache all packages from the configured content-

	shares. When the image is running in read-only mode you can deploy new packages or package updates. The agent will automatically use SCS mode (App-V) or AppAttach (MSIX) to make sure the write cache is not polluted.
Clear cache	This setting will instruct the agent to clear the cache at machine start before the packages from the content share(s) are pre-cached. Clear cache is an advanced feature: It will first remove packages and then remove any left overs in the cache to make sure it's empty. This setting is often used for non-persistent environments where the cache is redirected to another drive.

Refresh Cycle actions

Here you can configure the refresh cycle actions.

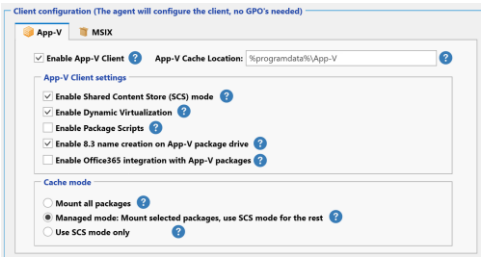


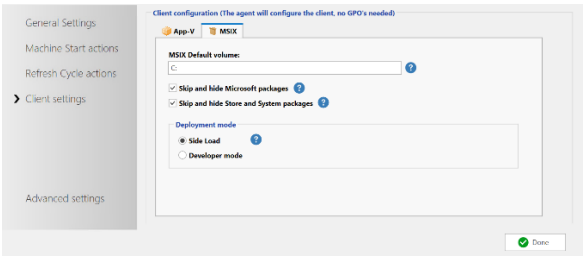
Refresh cycle actions	
Timer when the refresh cycle runs	You can configure the refresh cycle to run every x minutes. The refresh cycle already runs at machine start-up to perform cache management, you can enable a timer how often the refresh cycle should run during machine uptime and which part of the refresh cycle to invoke at the configured interval
Pre-cache packages from the content shares with the pre-cache option enabled	This is a mandatory setting, you can control if you want to pre-cache packages using the checkbox next to the content share in the machine group configuration window.
Execute user publishing tasks for currently logged in users	With this option you can publish new or updated packages to users while they are logged in. They do not have to logoff and back on again.
Only process publishing tasks for packages on	Publishing tasks are executed for users on machines regardless of where the package is stored, so a package could be added to a

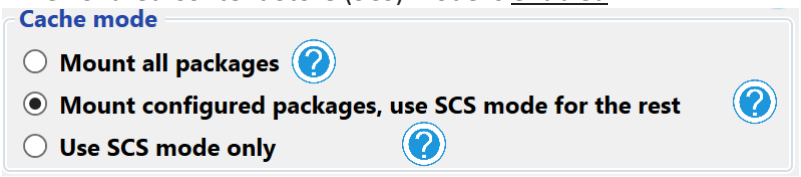
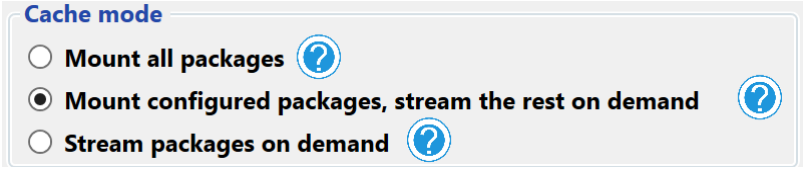
content share(s) configured for the machine group	machine from a content share which has not been configured for a machine group. With this setting only publishing tasks are executed for packages that are on one of the content share(s) configured for the machine group.
Remove packages from the cache that are no longer on the configured content share(s)	This option will remove packages from the cache when they are no longer found on one of the configured content share(s). This settings is especially handy for persistent environments to keep the cache clean and up to date. This setting is often not needed when you enabled the clear cache at machine start action.
Process Deployment configuration files	When this setting is enabled (default), the AppVentiX agent will process the deployment configuration file that's in the same folder as the package. The configuration file needs to have the .appd extension instead of .xml. This makes sure only configuration files are processed that are meant to be processed, without deploying all configuration files by default. (Applies to App-V only) Multiple .appd files in the same directory are supported, when you want to assign a specific .appd file to a package, provide the .appd file the exact name of the package (for example when package name is Archi_530.100.appv give the deployment config file the name: **Archi_530.100.appd** or **Archi_530.100 _DeploymentConfig.appd**)
Process global pending tasks	When a global published package is in use when a newer version is deployed the App-V client will generate a pending task to publish the package when the machine reboots. This is often not desired and the package needs to be available without the users logging off and having to reboot the machine. With this setting enabled the AppVentiX agent will detect global pending tasks and process them automatically when the package is no longer in use. The user only have to close the application and the agent will publish the new version immediately when it's no longer in use without a machine reboot. (Applies to App-V only)
Remove user pending tasks	Remove user pending tasks before executing user publishing tasks to make sure the publish operation is retried when a package was in use the previous time the publishing operation was executed (Applies to App-V only)
Enable registry prestaging	The pre-stage virtual registry option makes sure the virtual registry of the package is already loaded on the machine directly after the package is added (this is normally done when the user starts the application for the first time which can cause launch delays). This feature is especially for bigger packages in combination with non-persistent environments. You need to configure a service account if you want to use this feature.
Enable draining	Enable this option if you want to use the draining feature. When selecting packages in Central View you can select "Drain this package" in the package options. The agent will remove packages that are on the drain list and will prevent them from deploying again.

Client settings

With client settings you can configure the App-V and\or MSIX client. You don't have to configure any GPO's, the agent will configure the settings for you. A single point of configuration.



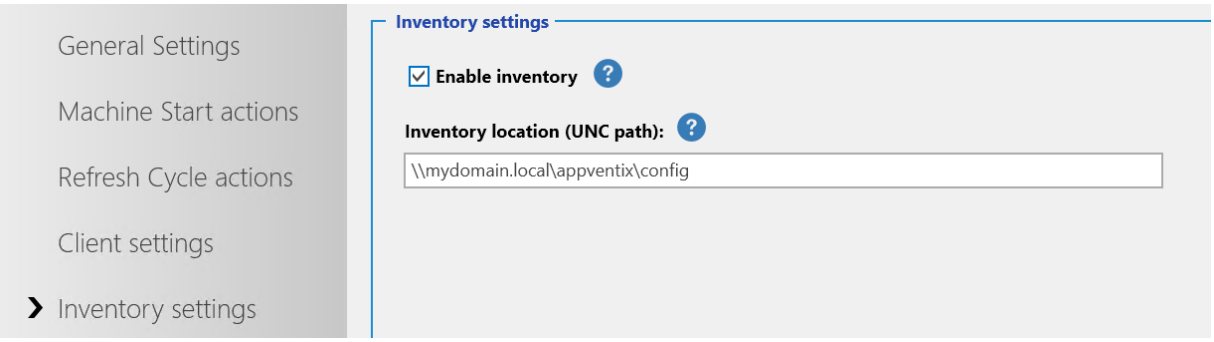


Client settings	
Enable App-V Client	The AppVentiX agent will enable the App-V client for you, no need to enable the App-V client on the machines manually
Cache location	The location where the package cache is stored
Enable SCS mode	Here you can configure if SCS mode should be enabled or disabled (it will be enabled by default). The service will configure SCS mode automatically when the service starts. (Applies to App-V Only)
Enable dynamic virtualization	This setting is default on but you might check to turn this off, this feature will integrate packages automatically with internet explorer and windows explorer (for example for plugins and context menus in explorer). If you want to have more isolation and control turn this setting off. (Applies to App-V only)
Enable package scripts	This setting is default off, if you enable this setting the App-V client will allow the use of package scripts. (Applies to App-V only)
Enable 8.3 name creation	The App-V client requires 8.3 name creation to be enabled on the disk containing the packages, this option will make sure it is enabled
Enable Office 365 integration	With this setting you can enable virtual applications to be integrated with the installed Office 365 installation (applies to App-V only)
Cache mode	Here you can configure the desired cache mode, you can choose from the following options: When Shared Content Store (SCS) mode is enabled:  When Shared Content Store (SCS) mode is disabled:  - Mount all packages will always mount (pre-cached) all packages in the cache - Mount configured packages (this is the recommended cache mode) (you can configure packages that should be mounted (pre-cached) in the cache in the Central View console. For other packages SCS mode is used or they are streamed to the cache when SCS mode is disabled - Use SCS mode only, no packages will be mounted (pre-

	cached) in the cache, every package uses SCS mode and reads the content directly from the network share - Stream packages on demand (when SCS mode is turned off) packages are loaded in the cache when they are started (applies to App-V only)
Deployment mode	Side Load is needed to deploy MSIX packages. This is now default in Windows. Please refer to the MSIX documentation for more information. (applies to MSIX only)

FSlogix App Masking feature	
Enable App Masking management	This feature can be enabled with only one checkbox, when enabled you can centrally manage FSlogix App Masking rules and assignments.

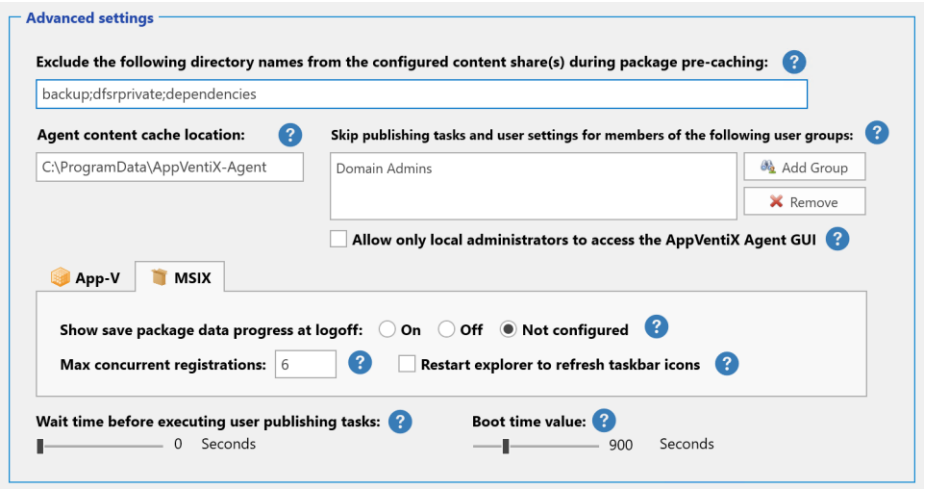
Inventory settings



In the inventory tab you can enable or disable the remote inventory feature, you can also configure another share to store the inventory data. By default the inventory data is stored on the configuration share. Please note that the machines need write permissions to the inventory location to be able to save the inventory data. The configuration share only needs read permissions.

Advanced settings

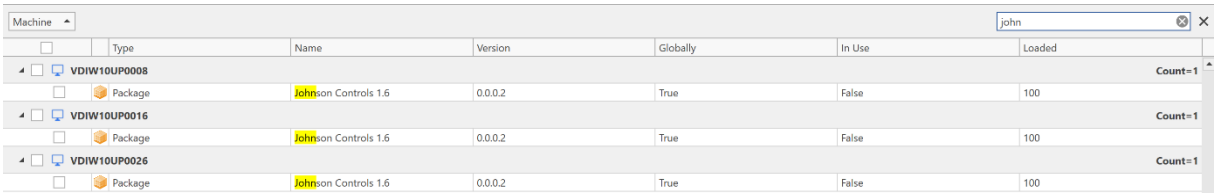
Only change advanced settings when you have a special use case for it.



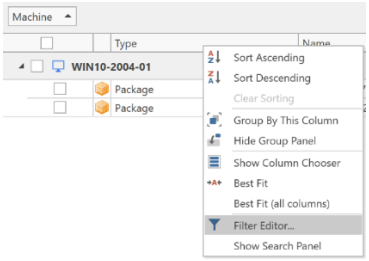
Advanced settings	
Exclude the following directory names on the configured content share(s)	A list of folder names that are skipped by the agent when performing pre-cache or balance cache procedure. By default a folder containing the name backup and dfsrprivate is excluded. The list is separated with a semicolon (;)
Wait time before executing user publishing tasks	Normally not needed, but here you can configure a specific timeout after a user login before the user publishing tasks are executed
Boot time value	The time the service detects that the machine has just rebooted and machine start-up actions are executed. By default this is 900 seconds. Increase this value if your machines have a very long start-up time.
Enable staged package load procedure (App-V)	In some cases, when a lot of packages are loaded in the cache at the same time, the App-V client doesn't load some packages correctly. When this feature is enabled, packages will be loaded using an interval to prevent this issue from occurring.
Enable App-V Cache validation after loading packages	This feature will validate the App-V cache and reload packages that are not loaded in the cache correctly. This feature makes sure the App-V cache is always healthy.
Show save package data progress at logoff	When user logs off the package data is saved, this will provide a status window to the user. It only takes milli seconds, so most of the time enabling this option is not needed
Agent content cache location	For the pre-cache and balance cache methods, the content share is scanned for changes. The content cache keeps a cache of the content on the share(s) so the number of reads is reduced. In non-persistent scenario's this location can be configured to a persistent drive like D:\AppVentiXCache
Max concurrent registrations	The number of parallel MSIX registrations at the same time (throttling), can be increased when using fast network\storage resources and can be reduced when having lower resources.
Restart explorer to refresh taskbar icons	In profile container scenarios with dynamic delivery (no pre-caching), pinned taskbar icons may appear transparent. This setting restarts Explorer once after login to refresh the icons on the taskbar.
Skip publishing tasks and user settings for members of the following user groups	Multiple groups can be configured (domain admins is added by default but can be removed). Users which are member of one of these groups will be skipped from user settings and publishing tasks.

Manage Machines page

The manage machines page in Central View allows you to inventory machines, easily check which packages are deployed on a machine. Compare machines with each other (check if they have similar packages deployed). You can also search packages by name:



It’s also possible to right click on a column header and set a filter to easily find the packages you are looking for:



You can select multiple packages and then click on Remove Package, this will remove a package from the machine(s) in real-time. (in the selected items ribbon menu). You can also filter machines so you only see machines that are online, you will find this filter underneath the select machine group dropdown box.

Actions can be performed per machine (for example inventory or refresh) or per machine group (using the machine group actions ribbon menu).



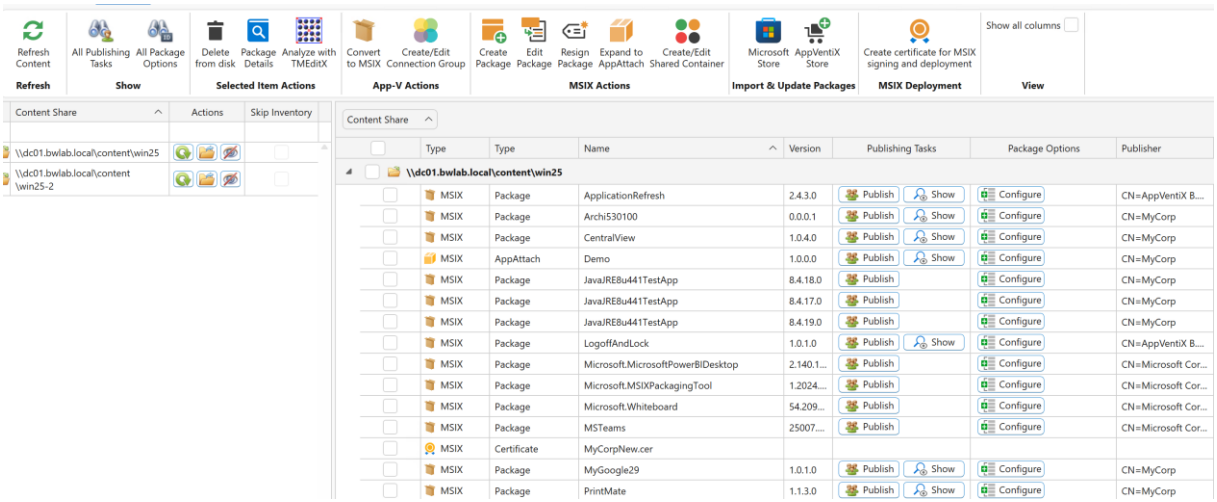
If you enabled both App-V and MSIX for a specific machine group you will be asked which feature (App-V or MSIX) you want to manage, you can switch the active feature with the dropdown box you will find in the ribbon menu. You can manage App-V and MSIX side by side.

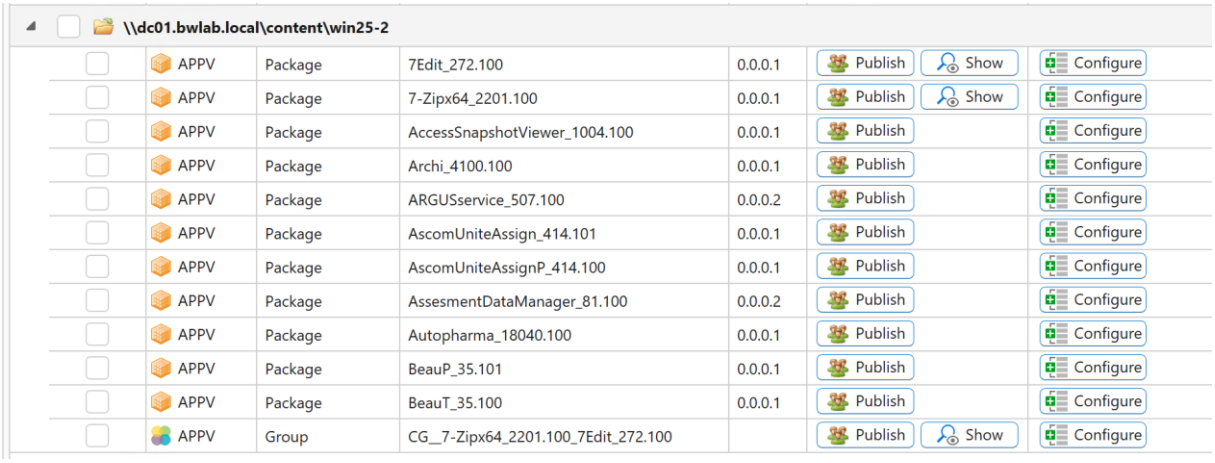
With the show virtual process button, you can inventory the virtual processes on machines and close them if needed (for example if you want to remove a package that is in use).

With the user inventory button you can see in real-time which users are logged in and which packages they have published. You can also invoke a repair from there.

The event inventory will allow you to see all the latest events from the agent.

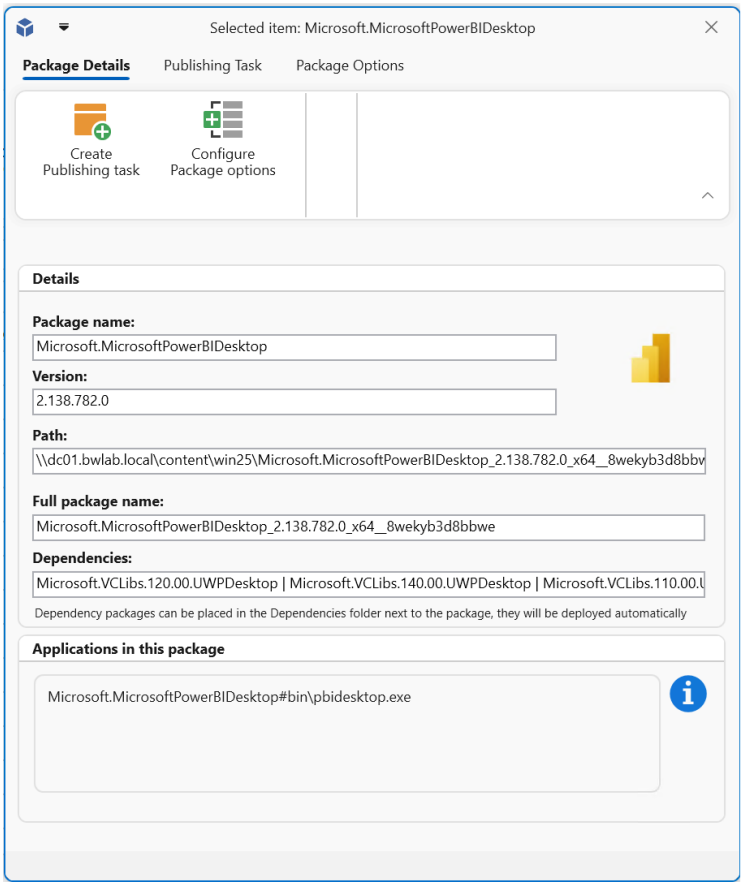
Manage Content page





The manage content page allows you to inventory the content share(s) and manage the central configuration. Click on the eye icon next to the content share to inventory the share. Or click on the refresh Content button in the top ribbon. Here you will find a lot of information about the packages, do not forget to select the Show All\Hidden columns checkbox to see even more information, like package size etc.

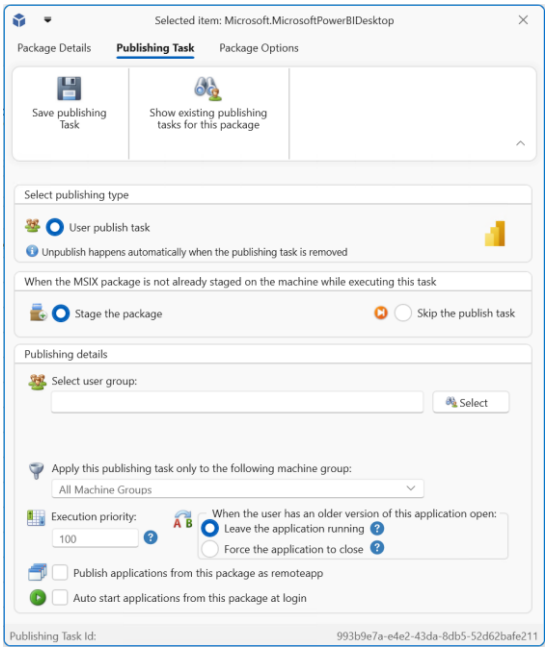
Double click on a package (or select the package and click on the Configure Package button in the selected item actions ribbon menu). To open the package options:



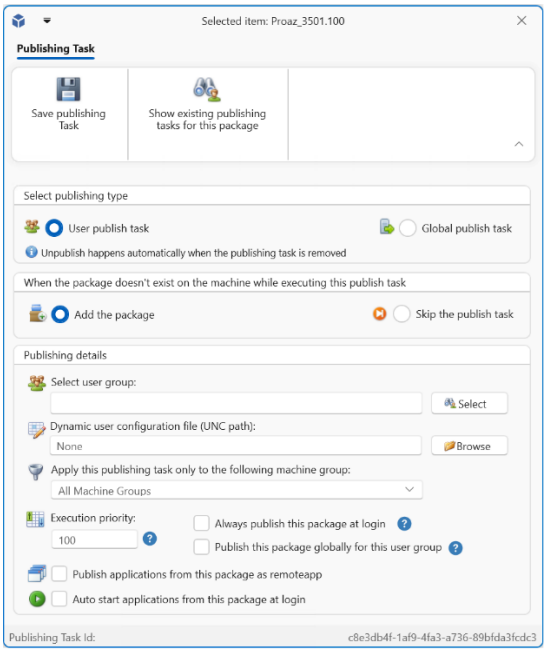
Here you can find details about the App-V or MSIX package and also create a publishing task or configure package options for the selected package. You can also click on the publish and options button next to the package it will take you to the new publishing task or package options window immediately.

Creating a publishing task

MSIX (app attach):



App-V:



When creating a publishing task, you can configure various options to suit your needs. Most options are straightforward. For instance, you can set the package to be published globally (making it available to everyone on the machine, applicable to App-V only) or restrict it to specific user groups (supported by both App-V and MSIX). There is also a publish to Everyone option for MSIX.

Unpublish tasks are unnecessary, as deployment is fully automated. Removing a publishing task will automatically unpublish the package for the user or machine.

You can define actions for cases where the package is missing during task execution. Options include dynamically adding the package or skipping the task altogether.

Using the machine group filter, you can restrict task execution to specific machine groups. By default, tasks apply to all machine groups.

Execution priority lets you control the order in which tasks are executed. Lower values indicate higher priority, which is useful if certain packages need to be published before others.

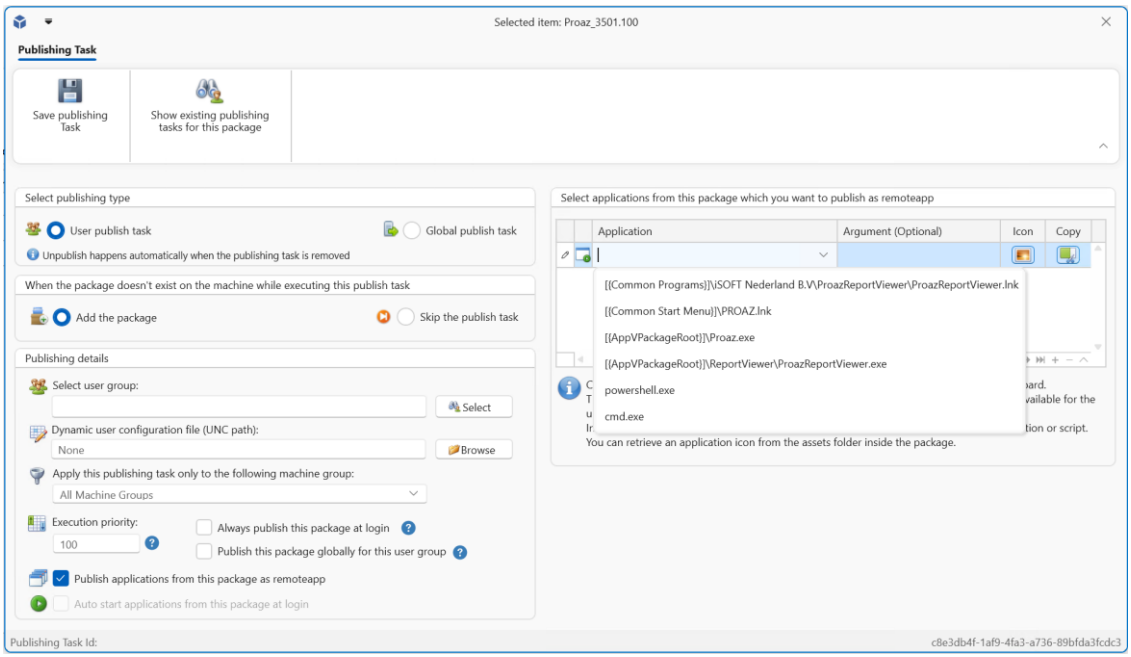
The agent checks for already-published packages and skips redundant tasks. However, the "Always publish this package even when already published" option overrides this behavior, ensuring the package is always published. Use this setting only if necessary, such as when dealing with profile solution issues (not typically required if the entire profile is maintained at logoff).

Some App-V packages, like Adobe Acrobat DC, must be published globally and do not support per-user publishing. The “Publish this package globally for this user group” setting allows the agent to globally publish packages based on user groups when required.

Desktop and Seamless application scenario

AppVentiX supports full desktop and seamless application publishing scenarios. Full desktop doesn’t need any additional configuration, just create and save the publishing task. For seamless application

publishing all you have to do is check the “Publish as remoteapp” checkbox. The window will be extended:



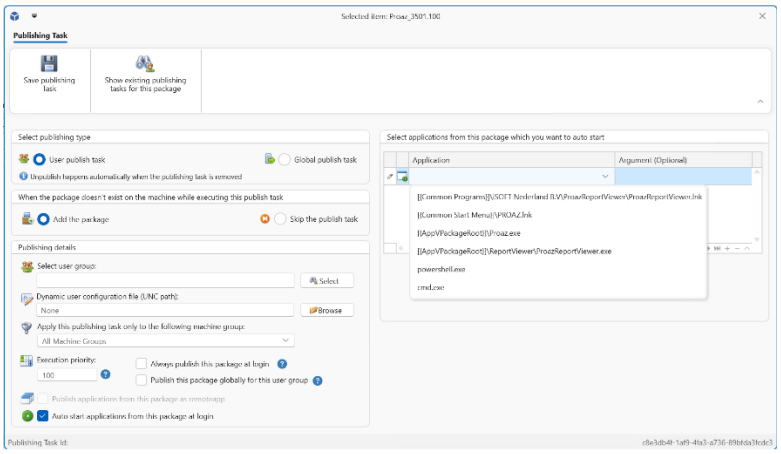
Now you can select the application you want to publish, or configure a powershell or cmd script to run. Scripts can run inside or outside the virtual environment.

When you click on the copy button next to the selected application the command to publish the application is copied to your clipboard, just paste this command in your seamless application delivery solution (RDS, Citrix, VMware, etc).

There is also a direct integration with Azure Virtual Desktop (AVD), this integration will publish the application directly in AVD, and also remove the application when the publishing task is removed. You can read more about this integration later in this guide.

> Note:
> In full desktop scenarios the publish as remoteapp checkbox is not needed.

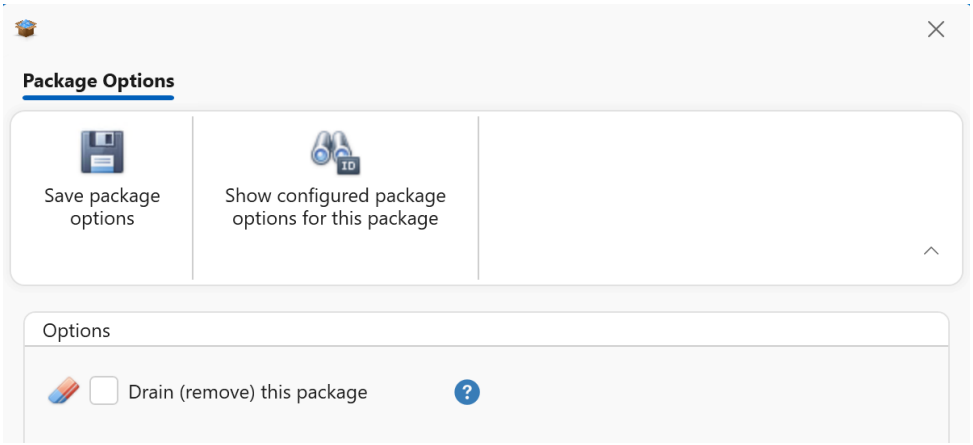
Auto start applications



With auto start applications you can select applications from the package that should be auto started for the user. The selected application(s) will be started automatically when the user logs in.

Configure package options

When you click on configure package options, you will see the following window:

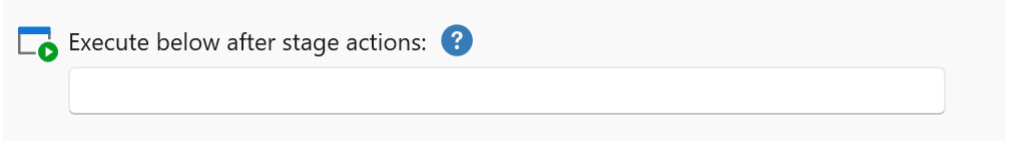


When you enable drain for a package, the agent will remove the package and prevent it from deploying again. Make sure that the enable drain feature is enabled in the agent settings (default). When there are publishing tasks configured for the package, they will be removed automatically.

Pre-stage registry (only applicable for App-V): use this option for very large packages, when enabled the agent will start the virtual environment of the package one time after it's deployed. This will increase the initial launch time of the application.

Machine group filter: You can filter on which machine group the configured package options should apply.

After stage actions



With after stage actions you can provide commands which are executed after the package has been staged on the machine.

For example when importing Microsoft Teams from the Import from Microsoft Store window, the following after stage action is configured automatically to install the Teams Office Addin:

msiexec.exe /i "C:\Program Files\WindowsApps\MSTeams_24004.1309.2689.2246_x64__8wekyb3d8bbwe\MicrosoftTeamsMeetingAddinInstaller.msi" ALLUSERS=1 /qn /norestart TARGETDIR="C:\Program Files (x86)\Microsoft\TeamsMeetingAddin";REG Add HKLM\SOFTWARE\Microsoft\Teams /v disableAutoUpdate /t REG_DWORD /d 1 /f

You can separate multiple command with the ; character.
Make sure to include quotes where necessary.

Show publishing tasks and Package Options

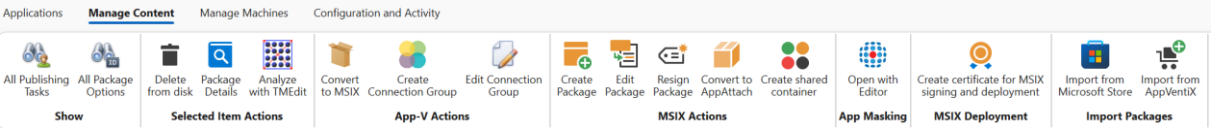
On the inventory content window, you will find two buttons:



When you click on Show publishing tasks you will see an overview of all publishing tasks. You can filter and sort the tasks, edit the tasks or delete the tasks.

The same goes for the package options, when you click on the Show Package options button you will see all configured package options.

In the selected items ribbon menu you will find different actions for selected items (depending on what package types are found in the inventory).



For example, you can create App-V connection groups by selecting specific packages or editing existing groups. Additionally, you can create MSIX shared containers directly from this page.

From the content page, you also have the option to convert App-V packages to MSIX, enabling a smooth migration at your own pace. Managing App-V and MSIX side by side is straightforward and efficient.

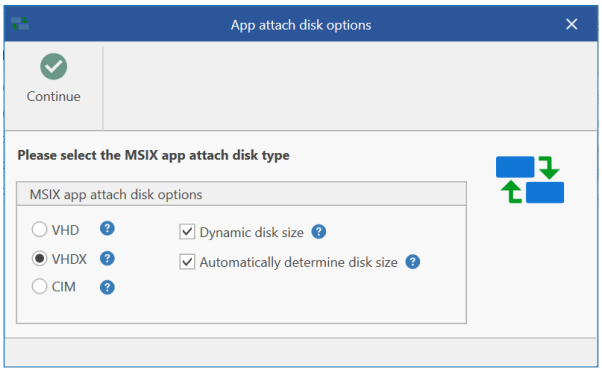
Please note you need a code sign certificate to sign MSIX packages, this certificate can be obtained from an internal or external Certificate Authority like Sectigo, or you can create one using the Central View console. When you create a certificate in Central View it will automatically be deployed to the machines so packages signed with this certificate are automatically trusted. More information about certificates can be found on page 30.

MSIX and MSIX app attach

With AppVentiX you can deploy normal MSIX packages and MSIX packages in a virtual disk format (called app attach mechanism). When you want to save disk space you can convert a MSIX package to app attach. This process will create an image file (VHD\VHDX or CIM) containing the extracted

MSIX package. This image will be attached by the AppVentiX agent so the application can be published to users. This delivery mechanism has the advantage that no disk space is used on the agent, but please note you can also just deploy MSIX packages without app attach, they will be cached on the hard disk of the machine instead of remotely attached. You can also combine to two delivery methods. After the package is converted to app attach it will be visible in the content inventory where you can publish it to a user group, the original MSIX package is stored in the backup folder and can be found there in case you need it for example to update the application. Because the original MSIX package is placed in a folder named backup the agent will not process\precache the original package.

Converting to app attach doesn't need any prerequisites, the convert process can run on Server OS (2022 or higher) and Client OS (Win 10 20H2 or higher or Win11). When you want to convert to CIM format, please note you need Win10 20H2 or higher, also please run the Central View console as admin to make sure the convert process runs smoothly.



The convert to app attach feature supports different options when creating the app attach disk, all disk options are supported (VHD, VHDX & CIM), you can choose dynamic disk size or a fixed disk size, the disk size is by default automatically determined according to the size of the MSIX package, but you can also configure the disk size manually.

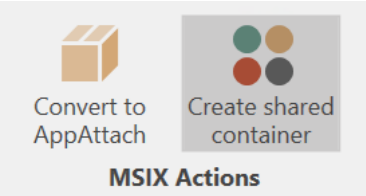
After converting a MSIX package to app attach it will be visible in the content inventory and you can publish it for users just like a normal MSIX package.
The AppVentiX agent takes care of all the actions needed for MSIX app attach (staging, de-staging, attach\detaching). It contains intelligent decisions and keeps track of the disks attached to the agent.

MSIX and app attach packages can be deployed and published to users even when they are not deployed\attached yet on the machine (on the fly delivery), making it a real dynamic delivery mechanism.

By default the registration for MSIX packages will be removed when the user logs off, this is done by the AppVentiX LogOff Handler (when persist MSIX app data agent setting is enabled). This process is needed when used in combination with FSlogix profile management and you want to roam the application data of the package between sessions. When you roam the profile of the user the persist MSIX app data setting is needed (enabled by default), when you have persistent machines per user (laptops etc) you can disable this setting. AppVentiX will take care of the whole process and supports both situations where the profile is roaming and non-roaming. Old (versions) of packages that you remove from the content share(s) will be removed from the agents automatically, making it a fully managed deployment solution.

MSIX Shared containers

MSIX Shared Containers are similar to App-V Connection groups, packages in a shared container will form one container so they can access each other's files and registry settings.



When you select multiple MSIX packages you can click on the "Create shared container", after you save the shared container it will be visible in the content inventory. You can then assign it to a user group and it will be enabled for the user automatically at the next refresh cycle or login.

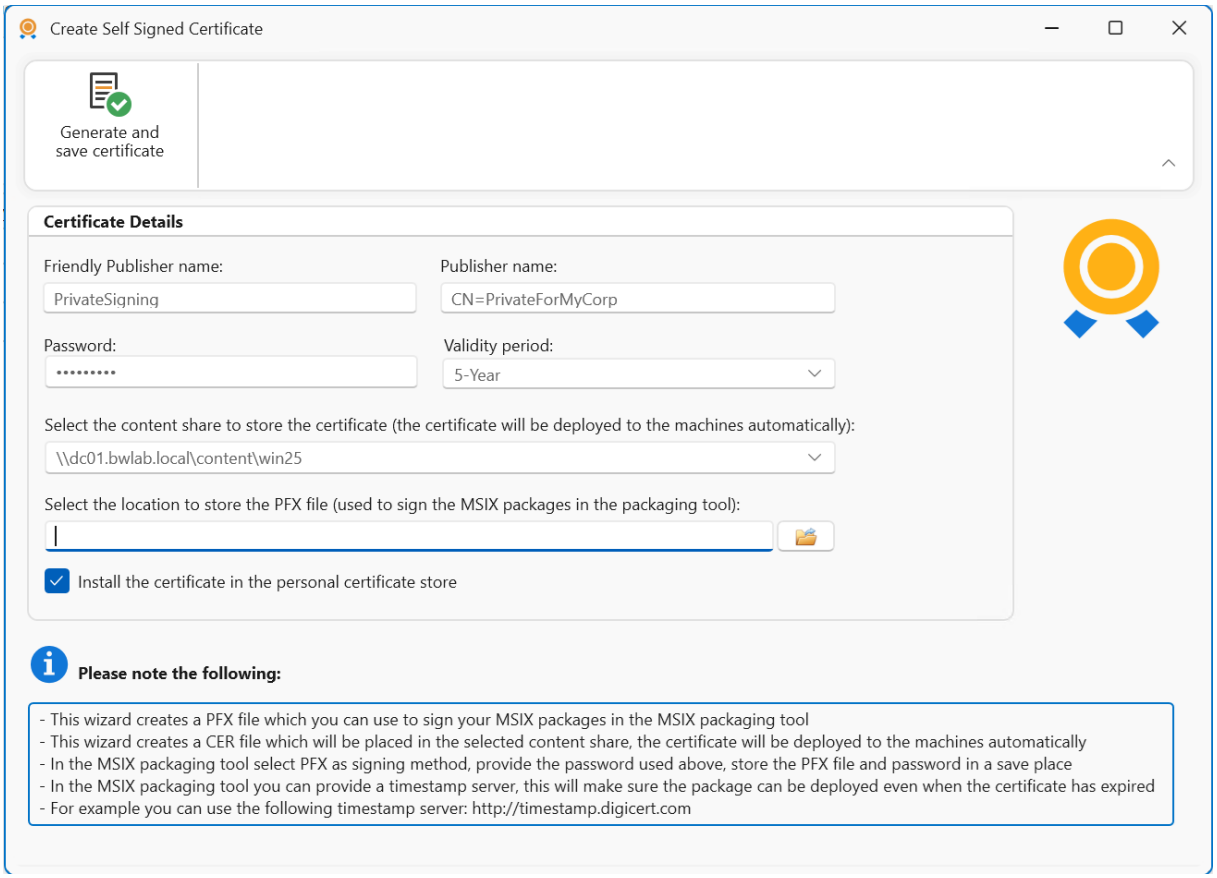
Please note that MSIX shared containers are only supported on Windows 10 Build 22x and higher and the current implementation of shared containers requires (local) admin permissions to apply the container.

MSIX Certificate management and deployment

Requesting 3th party certificates for signing MSIX packages is time consuming and costly. With AppVentiX it is now very easy to create your own certificate directly from the console. The certificate will be deployed to the machines automatically making it a very easy and quick method to sign and deploy MSIX packages.



When creating a certificate you can provide the following information:



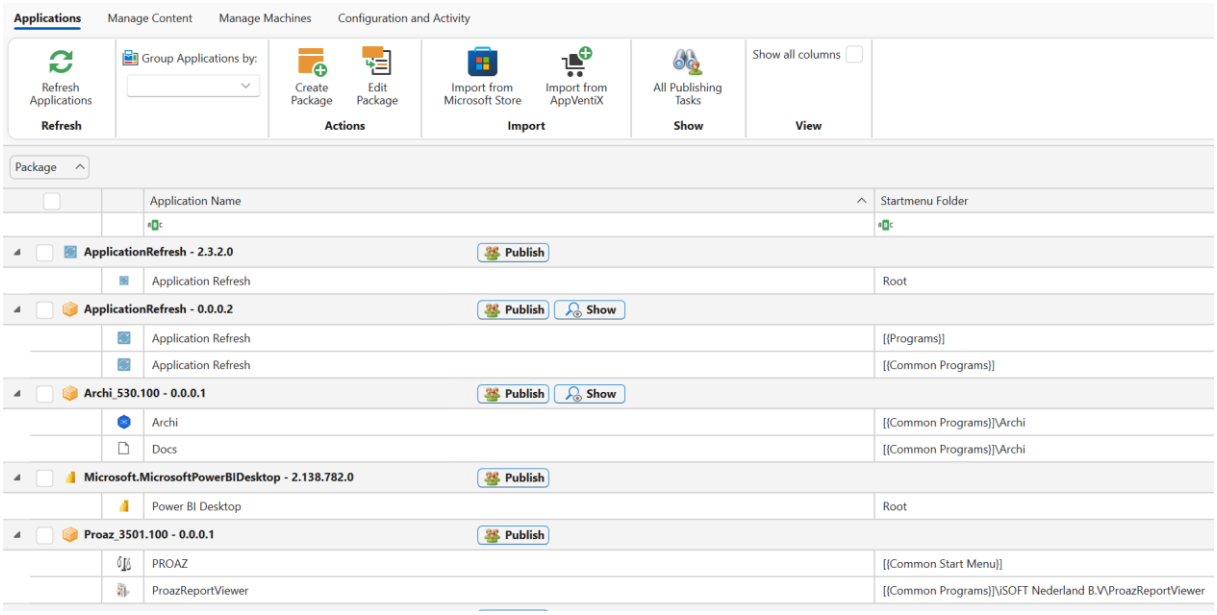
The certificate is stored as PFX and imported in the personal certificate store so it can be easily selected when converting packages from App-V to MSIX for example.

The .CER file will be stored on the content share you select and will be deployed to the machines automatically. The certificate is also visible in the content overview:



Application overview

The default page in Central View is the applications page:



This section provides an overview of all applications within your packages. You can easily create new packages or edit existing ones from here.

When creating new packages, you can manage shortcuts, include custom files, add scripts, and more, giving you complete control over your package configurations.

FSlogix App Masking management

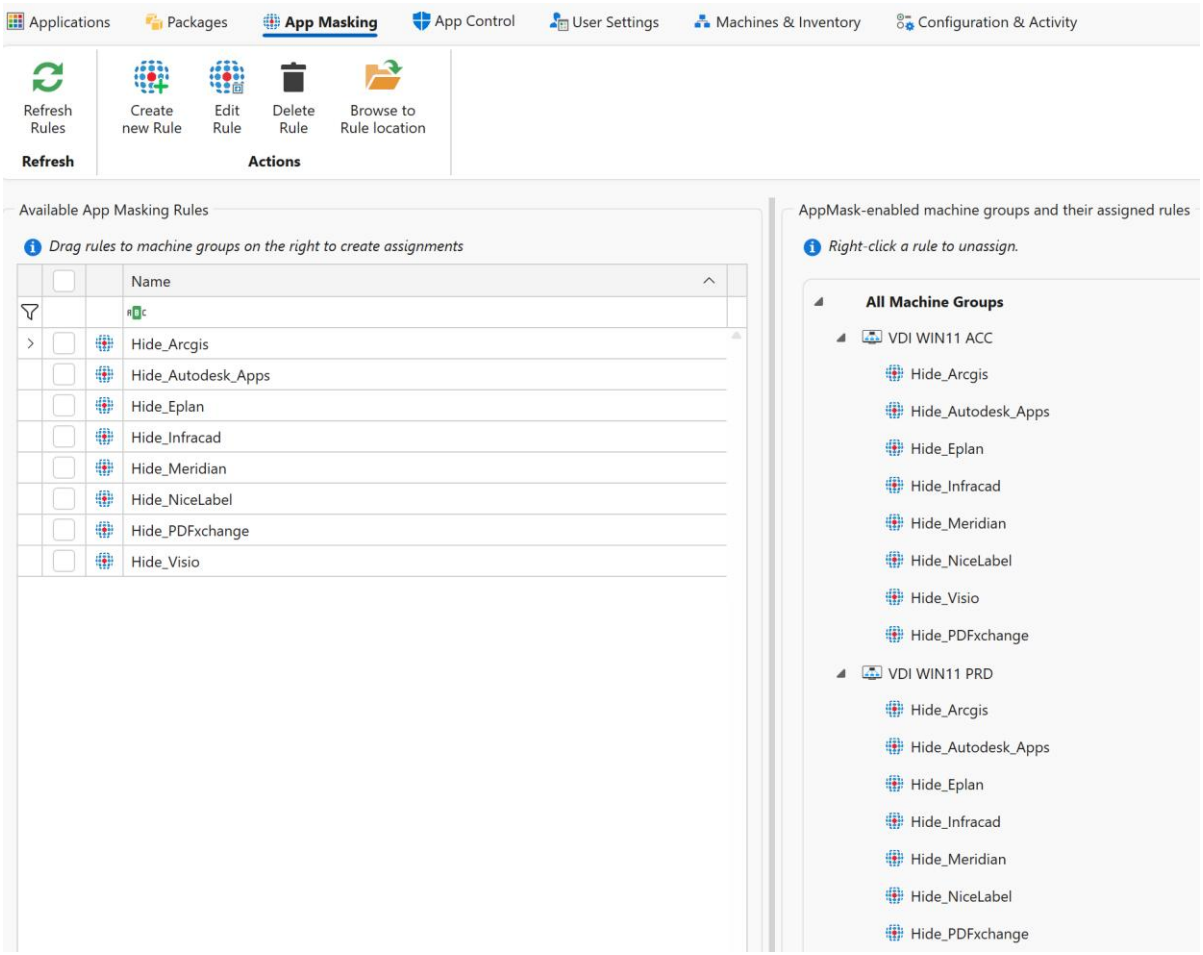
Software compatibility with App-V and MSIX is very high. However, there are scenarios where applications must be installed directly on machines without using App-V or MSIX.

With FSLogix App Masking, you can control which user groups can see specific applications by hiding files and registry items. AppVentiX makes it easy to centrally manage App Masking rules and assignments.

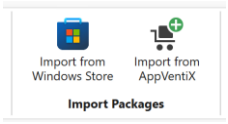
Click Browse to Rule Location and place your rules and assignments in that folder. Then click Refresh Rules and drag the rules to a machine group to deploy them to all machines in that group. When a rule is unassigned, it is automatically removed from the machines.

You can also modify App Masking rules and assignments directly from the Central View console. Changes are automatically updated on the machines. The Last Updated time in the machine inventory clearly shows which rule set and assignments are currently active.

Example of App Mask rules and assignments:



Import packages



Using the Import Packages buttons, you can directly import MSIX packages from the Microsoft Store, such as Power BI, WhatsApp, Whiteboard, and the MSIX Packaging Tool.

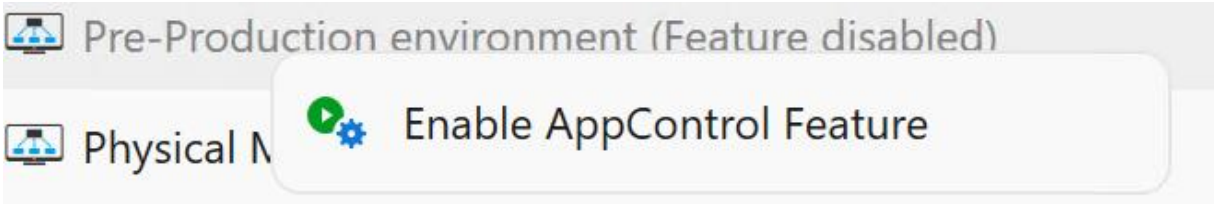
The Import from AppVentiX option allows you to import packages created by AppVentiX, including a test package for validating deployments, handy shortcuts like Log Off and Lock, and tools such as PrintMate, which lets users easily manage their printers.

App Control setup and best practices

AppVentiX makes it easy to setup and manage Microsoft App Control for Business (WDAC), this feature allows you to block applications and processes which are not allowed by your organization. This results in a very secure workspace because users can't run processes which are not covered in an allow policy.

Below the steps the get started with App Control quickly:

1: Enable the feature in the agent settings for the machine group or by right clicking on the machine group in the App Control page:



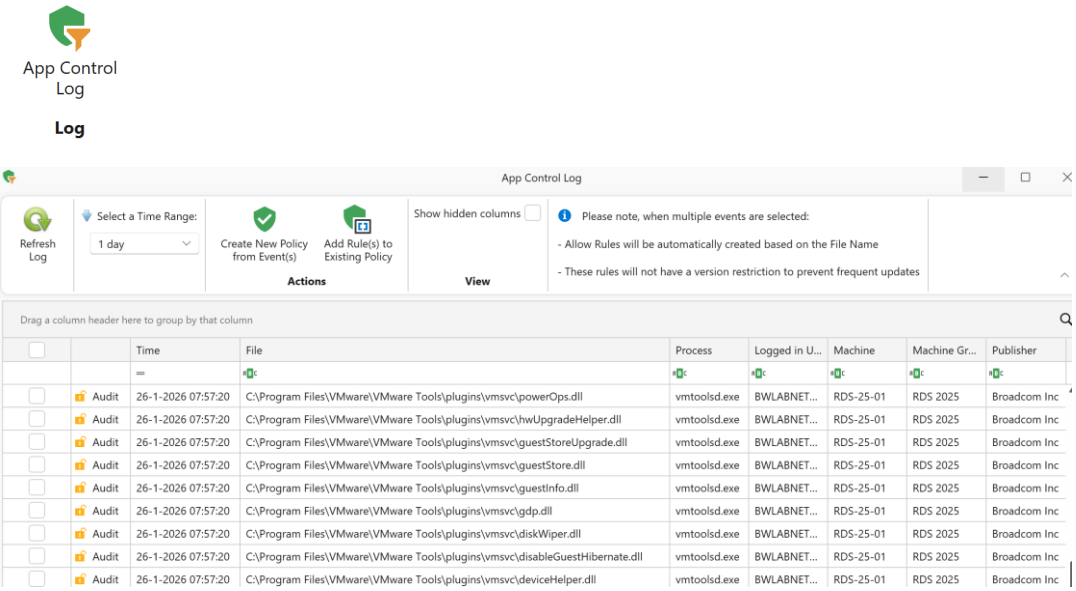
2: The machine group will be in learning mode by default:



After enabling the feature, restart the agent service or the machine to activate the feature.

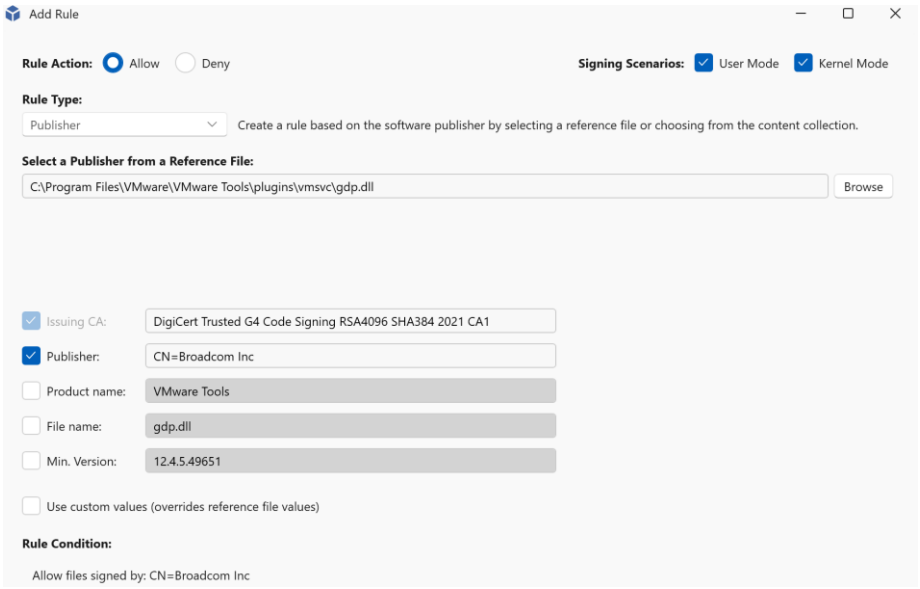
The learning mode will allow you to monitor the events so you can create policies to allow the applications and processes which you want to allow. At this stage nothing will be blocked.

3: Open the App Control log (note that it may take some time before the first events appear). You can speed this up by clicking the Refresh Workspace shortcut on the machine, which also initiates event collection.

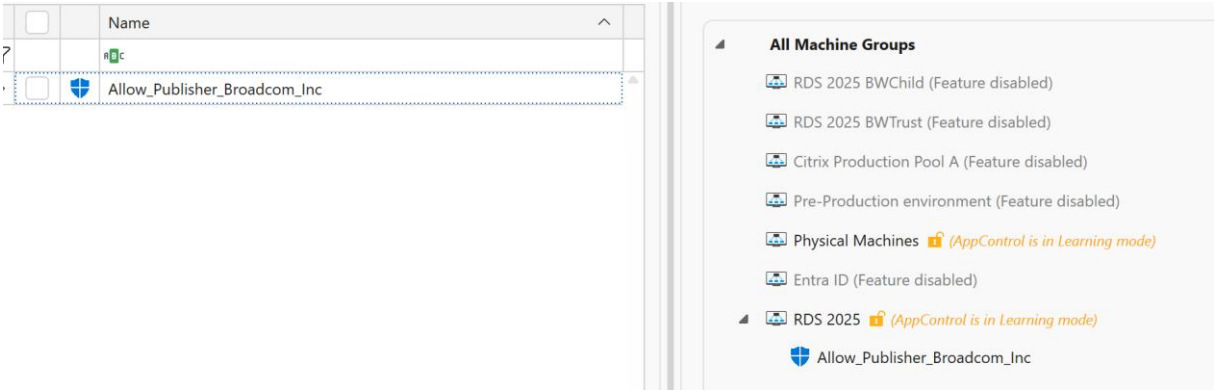


Select the event and click the "Create New Policy" button.

The recommended policy type is to allow the process or application by publisher (most applications are signed with a certificate). In this way you can create one policy to allow all the processes associated by this application. When application executables are not signed a file hash or file path rule can be used instead.

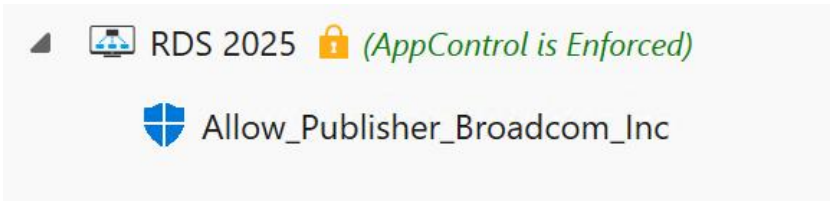


Save the policy and assign the policy to the machine group by dragging to the treeview:

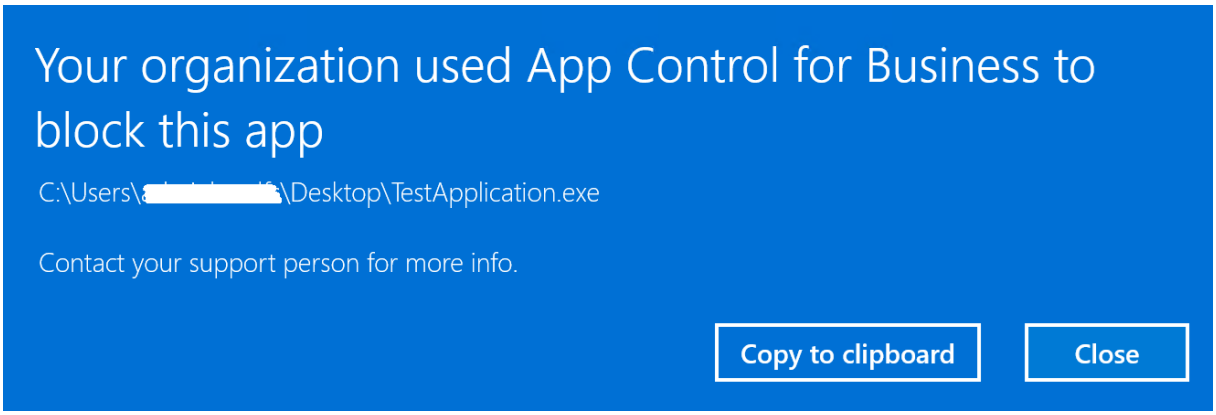


4: After the policies are created, right click the machine group and select “Enable Enforced mode”

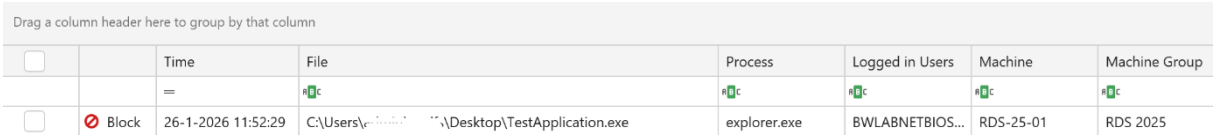




Applications and processes not covered by a policy will not be blocked and the following message is displayed to the user when an application is blocked:



Blocked events are visible in the App Control log window:



After reviewing the event, the process can be allowed in the same way as an audit event.When a new allow policy is created, the user can click the Refresh Workspace shortcut and immediately start the application, without needing to log off and back on.

Notes and best practices for creating App Control policies:

- Applications or processes that are allowed are no longer logged in the App Control log
- Creating policies that allow files or paths on UNC locations is supported only on Windows 11 and Windows Server 2025
- Publisher policies may not apply to service executables (processes launched by background services). In these cases, use folder path–based allow policies instead
- Policies that allow files in user-writable locations (for example, the user profile) require the Path is writable checkbox. This option is automatically enabled for profile paths. For other locations, verify that users have write permissions and enable this setting if necessary.



- Wildcard Usage: Folder path rules (C:\Program Files\MyApp*) apply to all files in the directory.
- With the Folder Scan rule type (creating a new policy from the App Control feature page), you can connect to a remote UNC share to scan a whole folder at once, it will provide you the publisher information and fallback to filepath or hash when files are not signed
- Creating policies from events is the easiest and most effective way to manage your App Control implementation

App Control Rule Types

App Control policies support multiple rule types with different specificity levels:

1. Publisher Rules: Allow applications based on code-signing certificate

- Most flexible and recommended for signed applications
- Supports publisher + product name + filename + version combinations

2. File Attribute Rules: Based on file metadata (filename, product name, file description, internal name, version)

- Useful for applications without valid signatures
- Less secure than publisher rules

3. File/Folder Path Rules: Allow/deny based on file or folder location

- Only works in User Mode (not for kernel drivers)
- Can specify individual files or entire directories with wildcards

4. File Hash Rules: Based on cryptographic file hash (SHA1/SHA256)

- Most specific but requires updates for each file version
- Fallback option when other methods aren't suitable

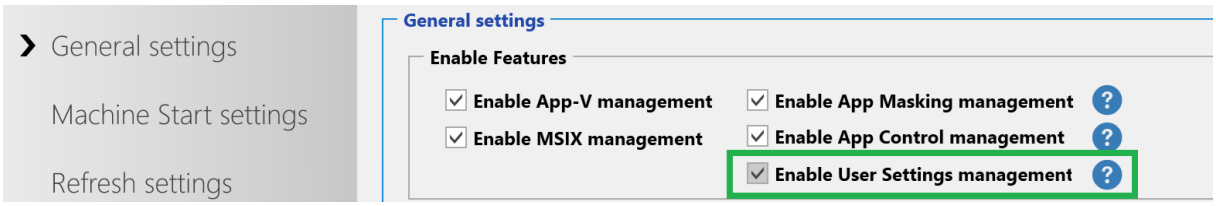
5. Folder Scan Rules: Scans entire folders and creates rules for all executables found

- Automatically creates rules using preferred rule levels (Publisher → FileName → FilePath → Hash)
- Supports excluding specific subdirectories

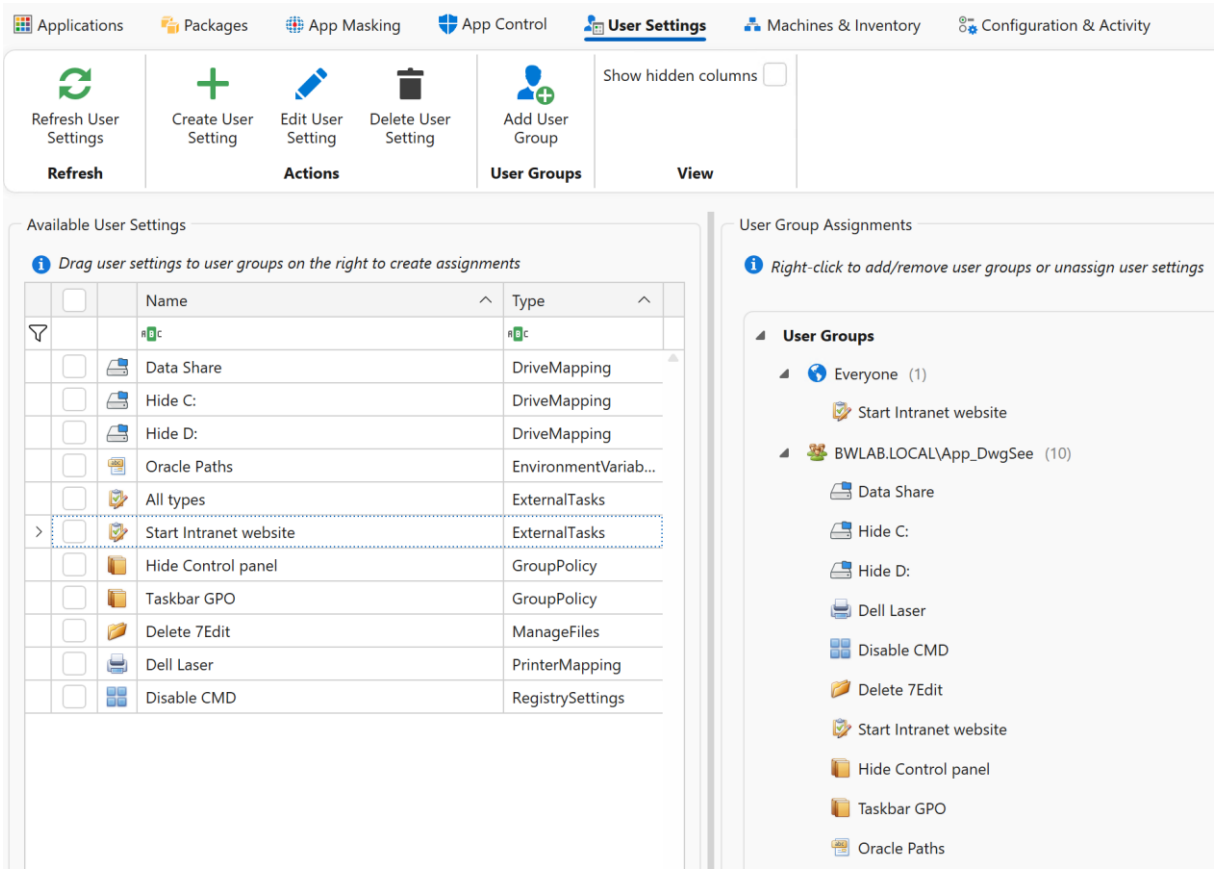
Configuring User Settings

User Settings allow you to manage the user workspace, including applying registry keys, drive mappings, printers, external tasks (with an embedded script editor), and group policies.

Usersettings can be enabled in the agent settings for the machine group:



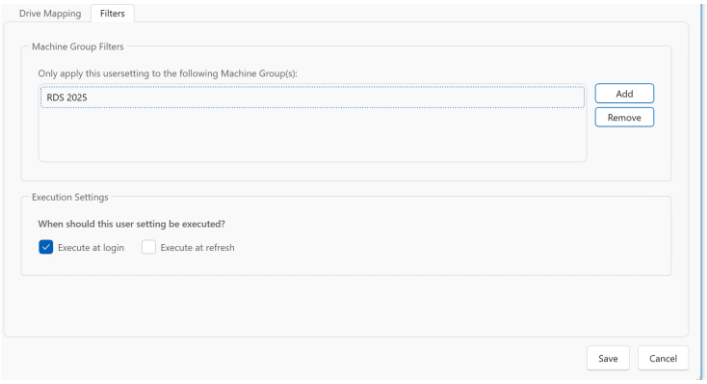
User settings can be created and assigned to user groups:



The everyone group is a built-in AppVentiX group which means the setting is applied to everyone.

Just create your user setting and drag it to the treeview to create assignments based on user groups. New user groups can be added with the “Add User Group”. Both Active Directory and Entra ID groups are supported.

In each user setting there is a Filters tab, in this tab you can narrow the user setting down to a specific machine group, and if the user setting should be executed only at login (default) or also when the Workspace of the user is refreshed.

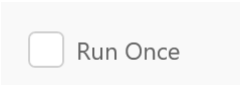


The following user setting types are tracked:

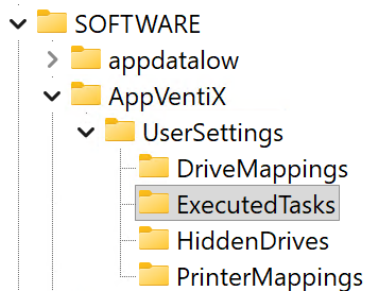
- Group policies
- Drive mappings
- Printer mappings

Tracked means that when an assignment no longer applies, the setting is automatically reverted. For example, if you remove or unassign a drive mapping from a user group, the drive mapping is automatically removed from the user. This ensures that no manual cleanup is required when changing drive mappings, printer mappings, or policies.

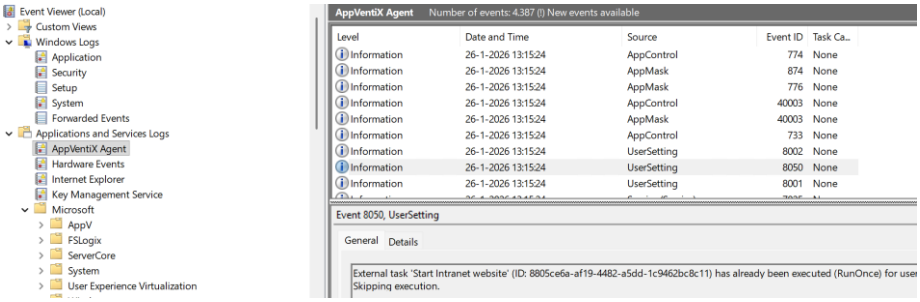
Some user settings have a RunOnce setting:



When enabled the user setting is applied\executed for the user one time, like copying a file or importing a regkey. When the user setting is applied, a registry key is created in the following AppVentiX regkey:

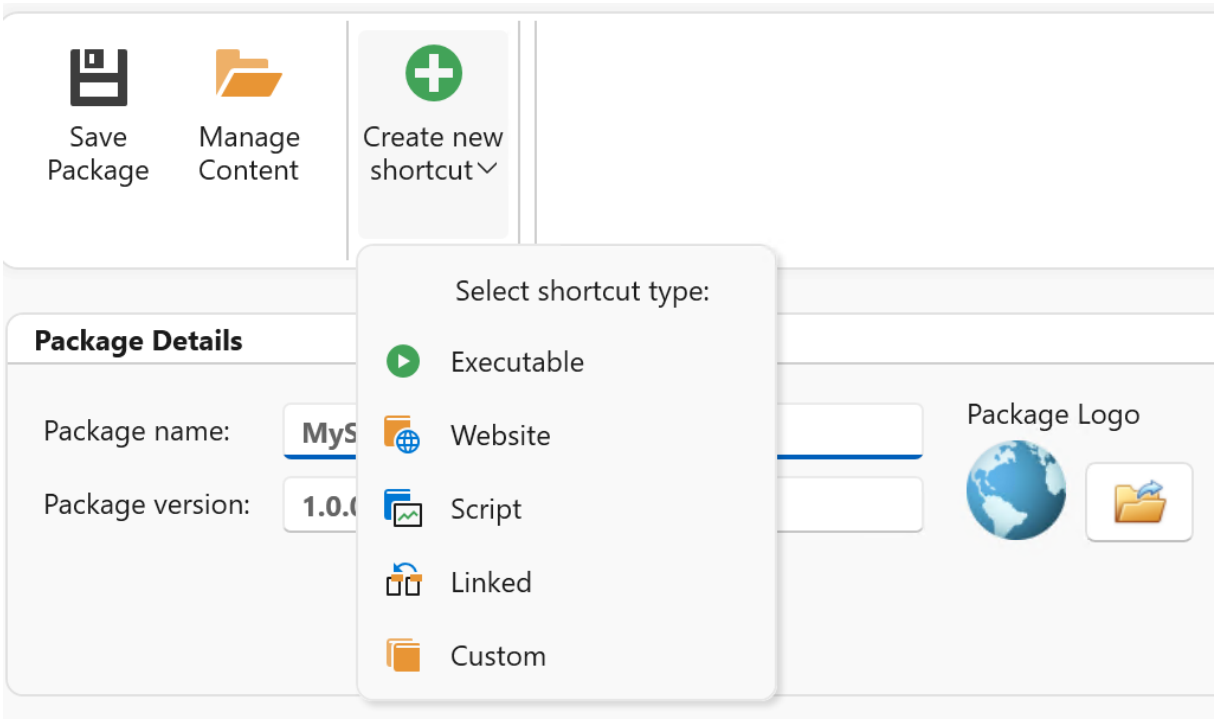


All user setting activities are also logged in the AppVentiX agent eventlog:



Creating shortcuts:

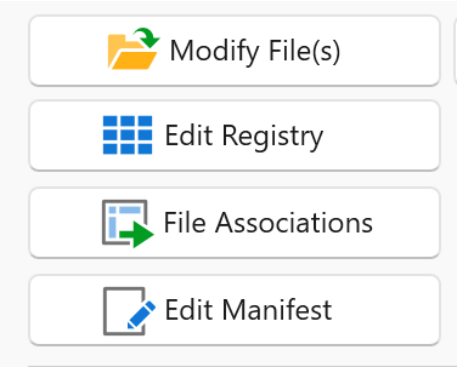
Leveraging MSIX it's very easy to manage shortcuts, you can even add (script) files to the shortcut package. Just create a new package and select the "Create new shortcut" button, you can start any executable (with or without arguments), this can be executables which you add to the package but they can also reside outside the package. For example you can configure Edge to start a specific website.



Example of a package containing shortcuts:



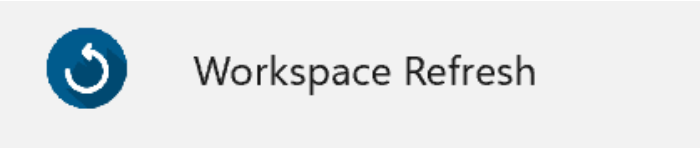
Using the manage content button you can optionally add files to the package:



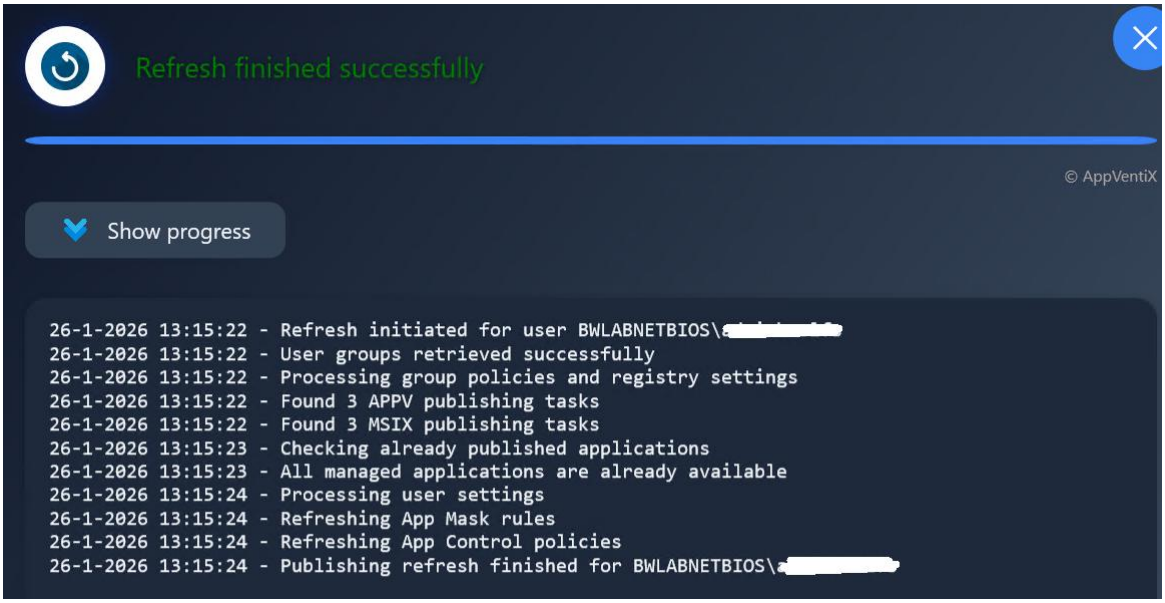
It’s also possible to configure FTA’s which point to files inside or outside the package.

Shortcut packages are assigned to user groups, in the same way as other applications.

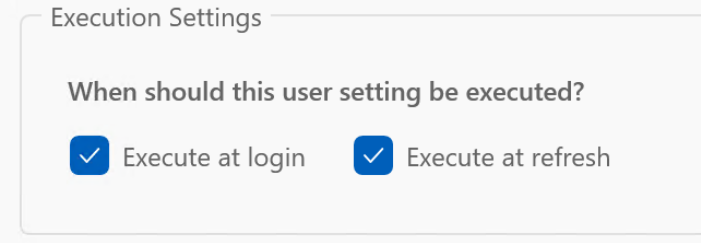
Refreshing the Workspace for the user can be done with the Workspace Refresh shortcut in the startmenu:



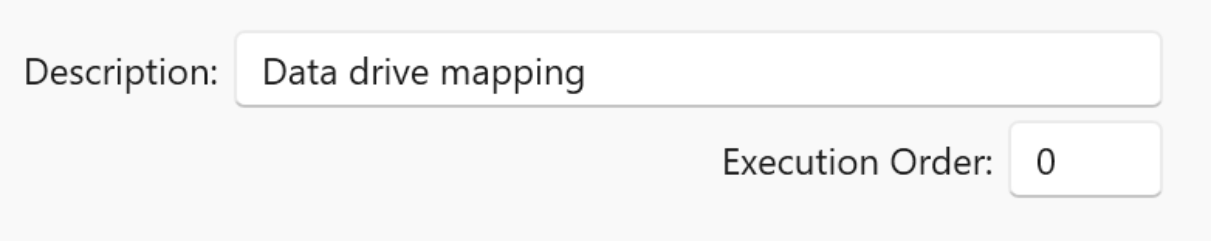
A progress will be displayed:



By default, User Settings are applied only at login. If you want them to also apply when a user clicks Refresh, enable this option in the Filters tab for the User Setting:

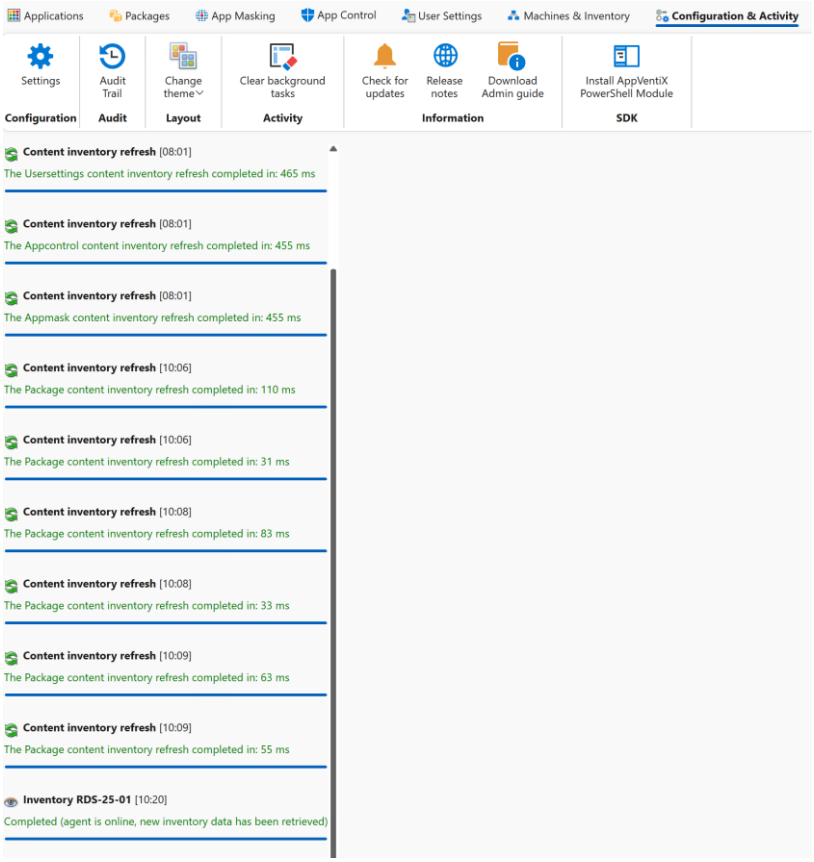


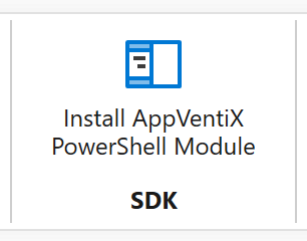
Execution Order lets you control the sequence in which User Settings are applied. Settings with a lower value have a higher priority and are applied earlier.



Configuration and Activity page

On this page you will find all activities from the Central View console, you can also change the Central View console configuration here or apply a different theme to the console.





A PowerShell module is available to automate the creation of publishing tasks. With above button you can install the module on the same machine as where Central View is installed. You can also install the module from the PowerShell gallery with below command:

Install-Module -Name AppVentiX

The configuration share will be automatically detected when the module runs on the same machine as Central View, if the module runs on another machine you can set the configuration share with below command:

Set-AppVentiXConfigShare -ConfigShare "\\path\appventix\config"

By invoking below command you will get examples how to use the module:

Get-Help New-AppVentiXPublishingTask -Examples
Get-Help Get-AppVentiXPublishingTask -Examples

Mandatory variables are: Type (MSIX\APPV), Group (the groupname) and Path (location to the package)

Firewall \ communication ports used by AppVentiX

By default AppVentiX doesn't use any other ports other than file share (SMB) access (445) or port 443 for QUIC enabled shares (read more about setting up a QUIC share in this admin guide).
There are a few exceptions:

- For AD domain connections (to retrieve user groups) port 389 (ldap) or 636 (ldaps) is used
- For Entra ID connections and Azure Virtual Desktop (AVD) connections the default Graph API ports (443) are used to Online Microsoft services
- When importing packages from the Microsoft store, the store URL needs to be accessible

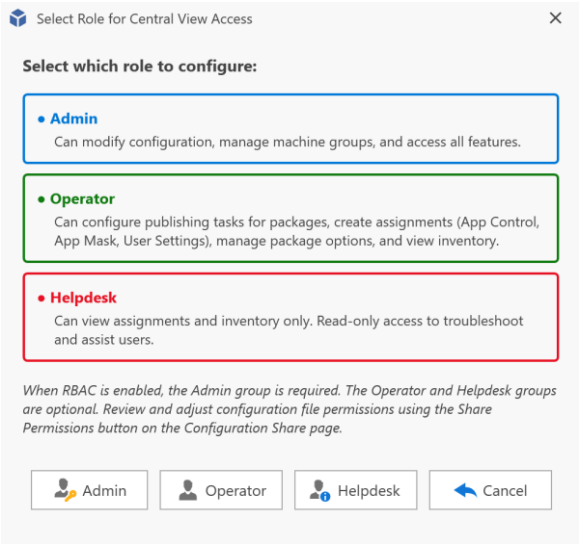
In Active Directory (AD) domain environments, the AD connection is most of the time already possible because a lot of authentication traffic is directed to domain controllers. AppVentiX will make use of this default AD integration already known in the operating system.

Limit access to the Central View console

You can configure an Active Directory group or Azure AD group that will have access to the Central View console. When a user starts the console the group membership is checked, when the user is not

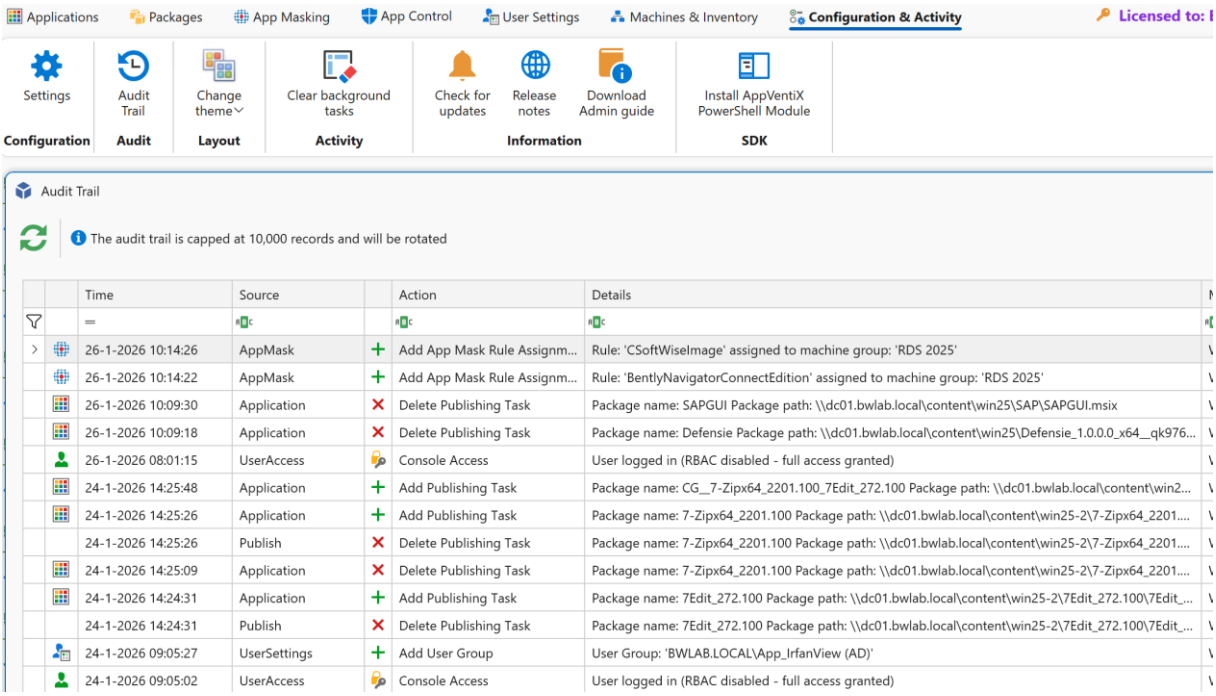
member of the configured group access to the console is not allowed. Additionally, you can configure other RBAC roles.

The following roles can be assigned to users or groups:



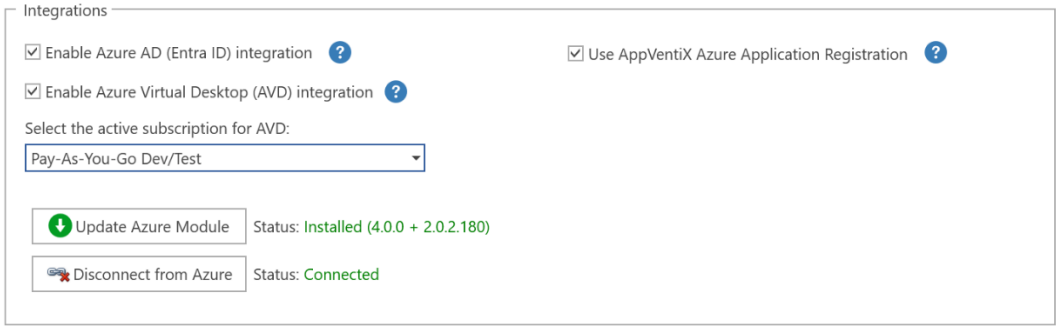
Audit trail

With the audit trail button you will get an overview of all the changes made to the configuration, the audit trail feature is enabled by default.



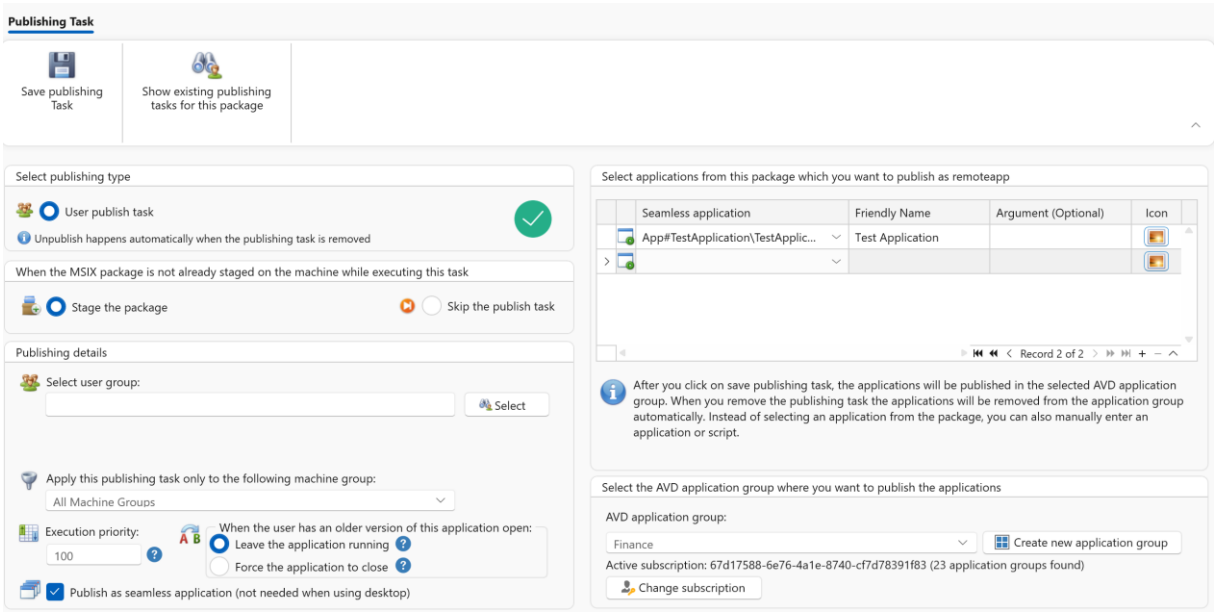
Azure Virtual Desktop (AVD) integration

AppVentiX supports both Desktop and RemoteApp scenarios. For AVD there is a direct integration available, you can enable this in the Central View settings:



Configuration is simple: Click on install AVD module and wait for the process to finish (this can take a few minutes), after installation click on connect to AVD and provide your credentials in the login box.

When the integration is enabled, you can select applications from an App-V, MSIX and\or MSIX app attach package and select an Application Group in AVD to publish the application to:



You can also create a new AVD application group directly from AppVentiX, you can also switch the active subscription and tenant if you have multiple AVD environments to manage.

When you save the publishing task, the applications are published in AVD and are accessible by the user, the AppVentiX wrapper will take care that the application is available for the user before it is started. It's also possible to launch powershell or cmd scripts.

It is also possible to edit a publishing task (like the group or application description), the applications will be updated in AVD automatically. When you remove a publishing task, the applications are automatically removed from AVD, making it a fully managed integration which is easy to use and provides complete control and insight from one single point of management.

The manage content page in the Central View console also includes a Remote App overview button:



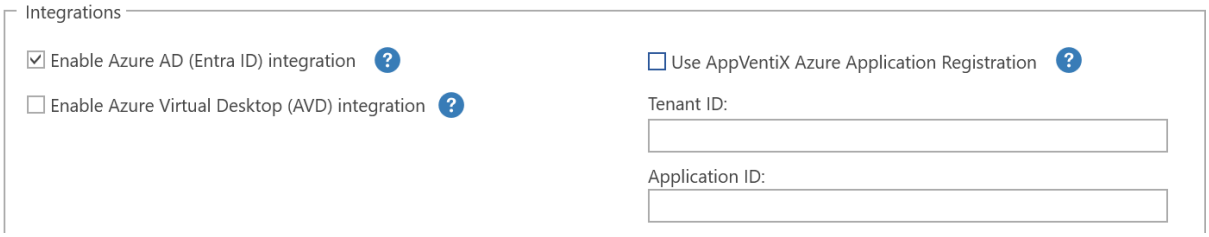
This will give you a total overview of all published applications in AVD, you can sort on application group, user group assignment and application name. This will provide you complete insight in one overview.

To manage application groups in AVD, you need at least contributor permissions in AVD:

Azure AD (Entra ID) integration

After you enabled the Entra ID integration in the Central View setting you have the option to use the AppVentiX Azure application registration or provide your own application registration. This is only used by the agent, not the Central View console. The Central View console uses the account you select to authenticate to Entra ID.

It's not needed to configure and provide secrets for AppVentiX, a basic application registration is enough for AppVentiX because AppVentiX will not connect to Entra ID by itself but instead uses the access token from the user using MSAL. This makes the solution secure because the secrets doesn't have to be stored by AppVentiX.

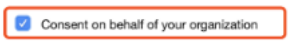


For Central View to be able to retrieve user groups and devices from Entra ID, the following Graph permissions needs to be assigned to the account you use to login to Entra ID:

- Device.Read.All
- Group.Read.All
- GroupMember.Read.All

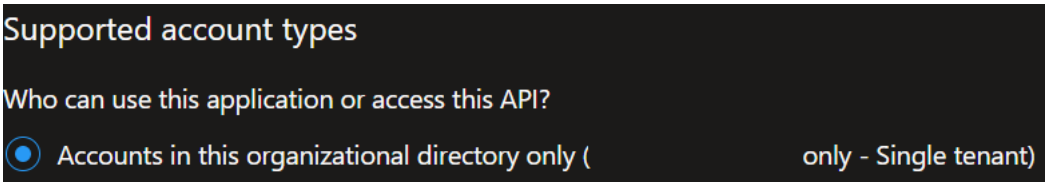
When you use the AppVentiX Azure Application registration:

The user will see a consent window after login, the only permissions requested are user.read to retrieve the group membership of the user. Recommended is to let an admin perform a consent one time (with the below checkbox checked) so users will not see the consent window at all.



When you configure your own Azure Application registration:

Create a new application registration in the Azure portal and give it a name like AppVentiX.

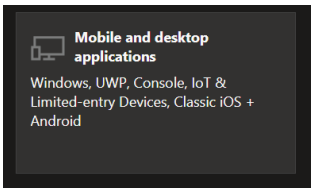


Choose this organization only.

You can enter the redirect url's later, click on save.

Go to the authentication menu in the application registration you just created.

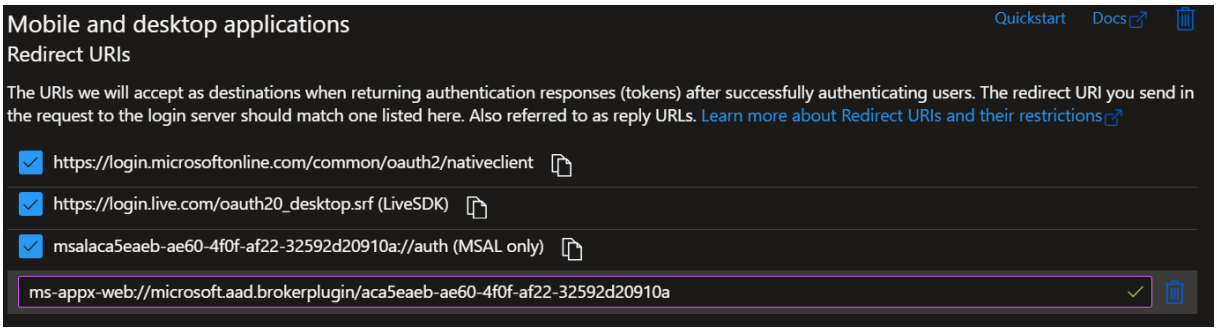
Click on add platform and choose mobile and desktop applications.



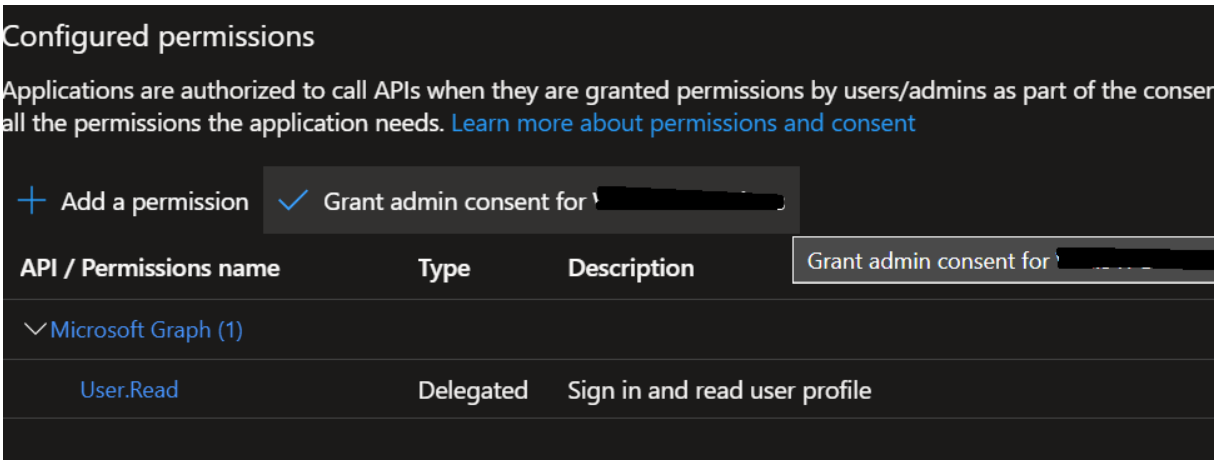
Enable the checkboxes and add one more redirect uri:

ms-appx-web://microsoft.aad.brokerplugin/aca5eaeb-ae60-4f0f-af22-32592d20910a

(replace the application id with your application id which can be found in the overview menu).
The end result looks like this:



Now go to the API permissions tab, the user.read permission should be already configured by default, if not add the user.read permission (more permissions are not needed!)

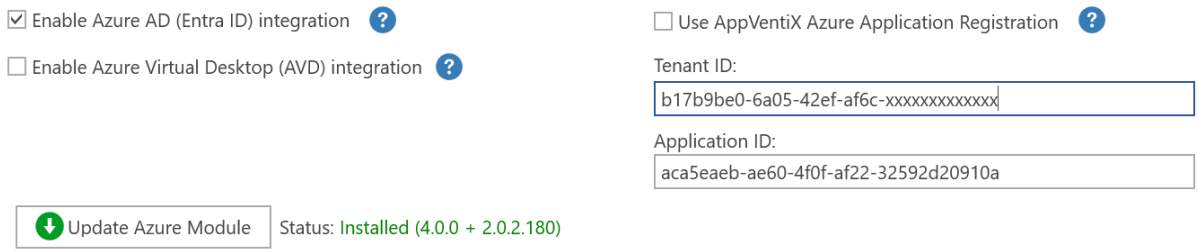


Click on Grant admin consent, so users are not prompted with the consent window.

Now go to the overview menu for the application registration and copy the application and tentant id:



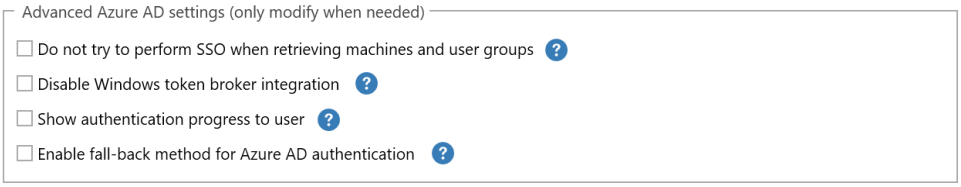
Paste them in the Central View settings window:



You are now finished, click on save.

Advanced Azure AD settings

There are some advanced Azure AD settings that can be configured, you only have to configure them when needed:



By default SSO will be used in the Central View console to retrieve machines and user groups, if you don't want to use SSO you can enable this setting.

The second checkbox is to disable the Windows token broker integration (WAM), in newer operating systems (Win10\11) the integrated authentication broker will be used by default. For Azure AD joined VMs this will work out of the box and normally you don't want to disable this.

Normally the authentication and Azure AD group retrieval happens silently in the background, with the "Show authentication progress to user" option the progress will always be shown to the user. This can be used for testing, but is normally not needed to enable.

Enable fall-back method for Azure AD authentication will use older Azure AD integration, only enable this option when MSAL can't be used.

Logging:

When a user sees a popup to consent or authenticate, the silent authentication did not work. You can find more information in the log file which is stored in the users profile:

C:\Users\username\AppData\Roaming\AppVentiX

AppVentiXLogonHandlerLog.txt

Hybrid Azure AD joined machines

For machines that are both Active directory integrated and Azure AD integrated (hybrid joined), use the machine groups based on AD group or OU.
Only use the Azure AD machine group for Azure AD (only) joined machines.
Azure AD joined (only) machines are not member of an Active Directory domain.

Azure AD joined (only) machines

Machines that are only joined to Azure AD, can be configured with an Azure file share which is stand-alone (not AD integrated). For this you can provide the storageaccount name and access key, please find more information in the Azure File share configuration section below.

Azure file share configuration

AppVentiX supports Azure file shares that are AD integrated and\or stand-alone shares.

Using stand-alone Azure file shares (for example for Azure AD joined machines):

In the Azure portal go to Storage Accounts and click on create storage account:



No storage accounts to display

Create a storage account to store up to 500TB of data in the cloud. Use a general-purpose storage account to store object data, use a NoSQL data store, define and use queues for message processing, and set up file shares in the cloud. Use the Blob storage account and the hot or cool access tiers to optimize your costs based on how frequently your object data is accessed.

Create storage account

Give the storage account a name and selection your region:

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription * Azure subscription 1

Resource group * (New) StorageAccounts

Create new

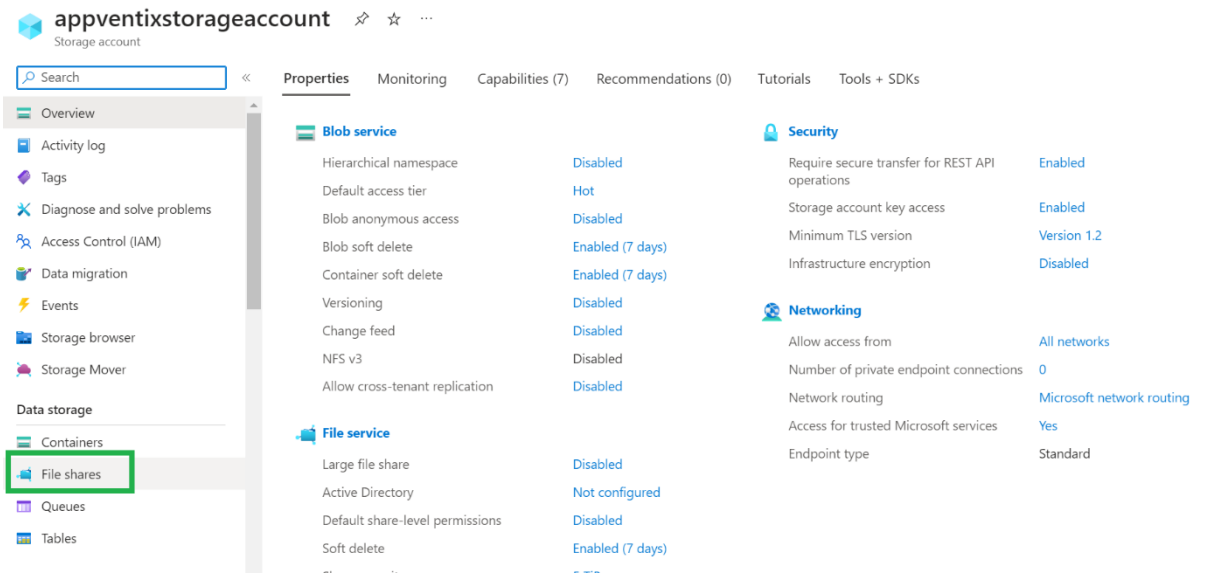
Instance details

Storage account name * appventixstorageaccount

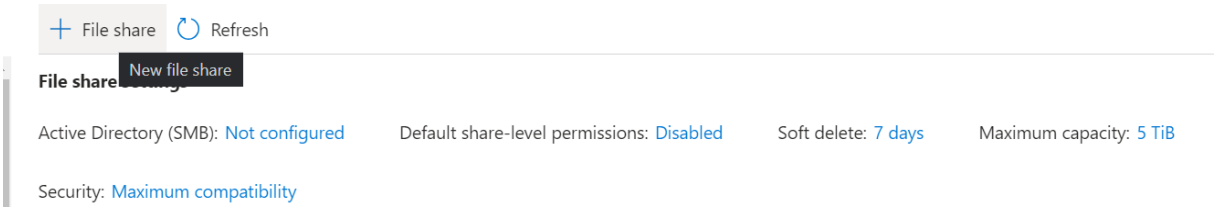
Region * (Europe) West Europe

You can leave all settings default and click on create. Optionally you can fine tune the storage account to meet your needs, but the default settings are already enough.

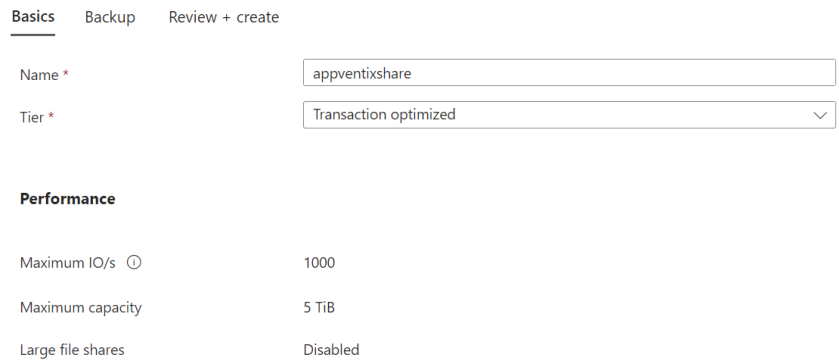
After the storage account is created we are going to create the file share.
Go to the storage account and click on file shares:



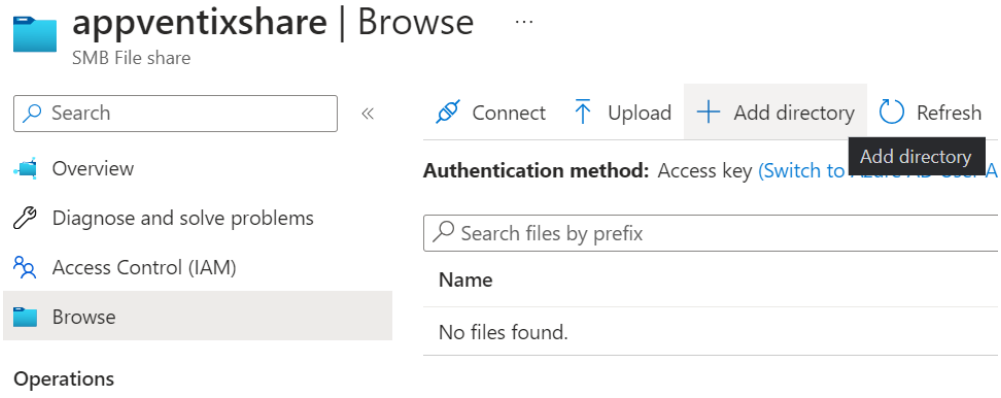
Select new file share:



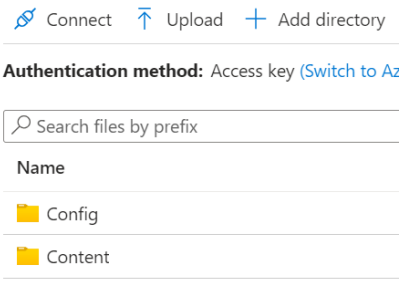
We will select the default tier and settings, they have been tested to work and perform well with AppVentiX. Optionally you can decide to pick a higher or lower tier, you can always change the tier later.



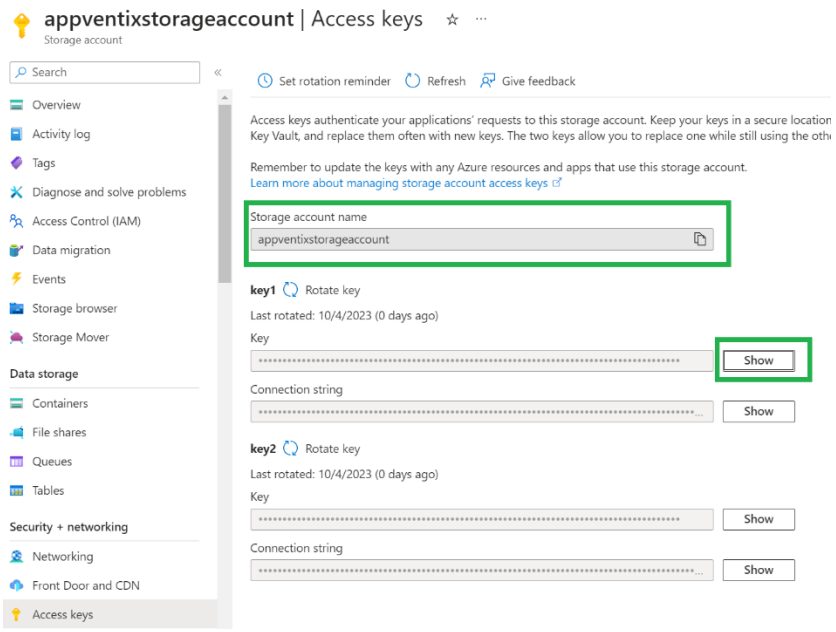
Click on create to create the file share.
After the file share is created, click on Browse and then Add directory:



Create 2 folders (config and content):



Go to the storage account main menu and click on Access keys:



From here copy the storage account name and the access key, now go to the Central View settings page.

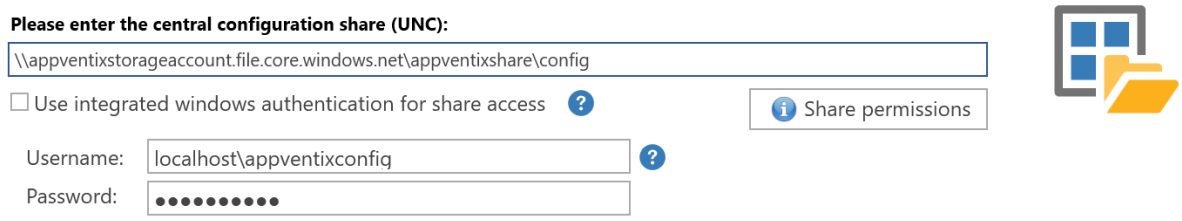
Enter the file share you just created in the below format (note the config folder at the end):

\\appventixstorageaccount.file.core.windows.net\appventixshare\config

(Replace the storage account name and share name with the one you just created)

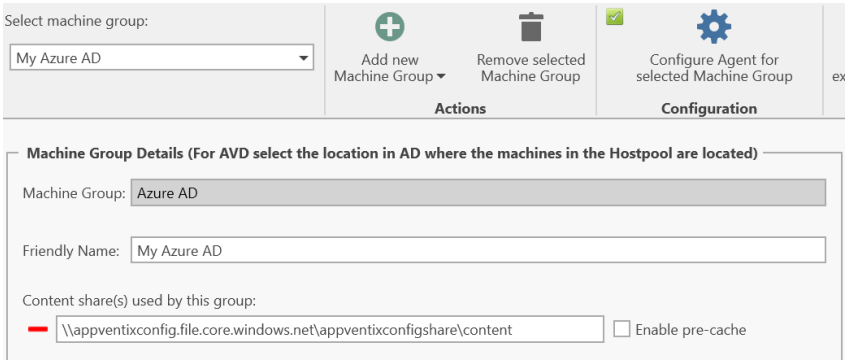
Clear the checkbox and copy the storage account name in the username textbox, make sure to use localhost\storageaccountname.
In the password field paste the access key.

Example:



Save the settings, if the settings are saved successfully it means the share is accessible and configured correctly.

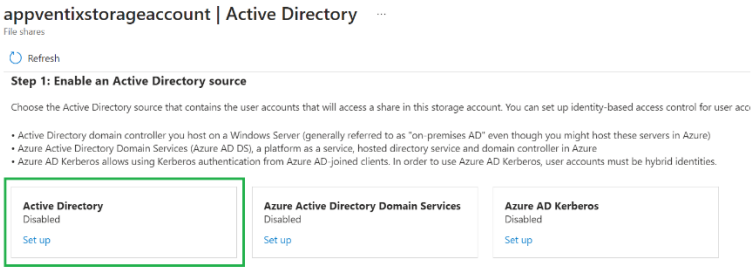
The content folder which you have created in the previous step, can be used in the machine group configuration:



This is the location where you can store the App-V, MSIX (app attach) and\or FSlogix app masking rules on. In the manage content page you can browse to the share and upload content.

Using Azure file shares that are AD integrated:

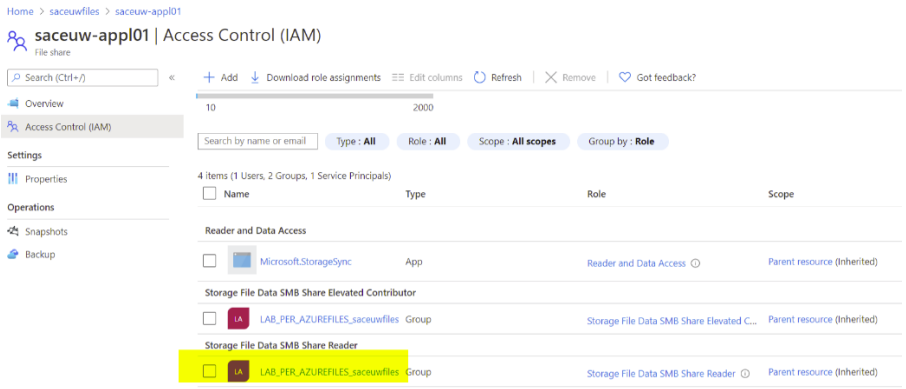
Use an existing file share or create an Azure file share just like in the previous steps, then join the share to Active Directory:



(note you can also use Azure AD Kerberos integrated, but the steps to configure the share in AppVentiX will be the same as the stand-alone share configuration).

After you completed the steps to integrate the share with your Active Directory, you have 2 options to authenticate to the share:

1: Create a group and add the machine accounts to the group, give this group read permissions:

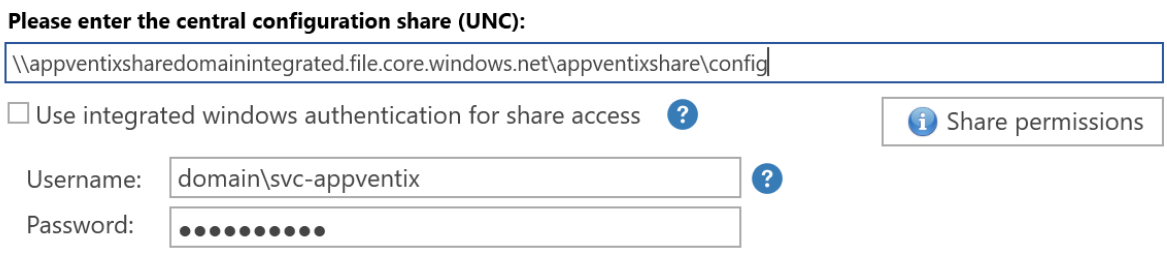


Make sure this group has read\write permissions in the inventory folder on the configuration share or else the remote inventory will not work. You can leave the integrated authentication checkbox enabled.

2: Create a service account and give this user account read\write permissions on the share



Then enter this account in the Central View settings window:



How to setup a QUIC share to allow share access over port 443

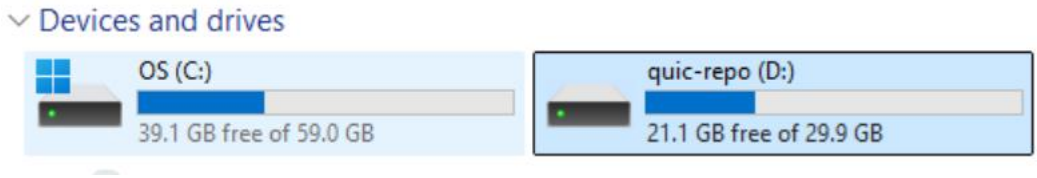
Default SMB uses port 445 which can be blocked by internet ISP's, there is a Microsoft solution for this called QUIC, this is SMB over HTTPS, using the highest TLS encryption available today. The server hosting the QUIC share can be placed in the DMZ, no Active Directory communication is needed.

QUIC can be enabled on an Azure VM using the admin center in the Azure portal, or manually by using for example Powershell. Below the steps how to setup the QUIC share with Powershell.

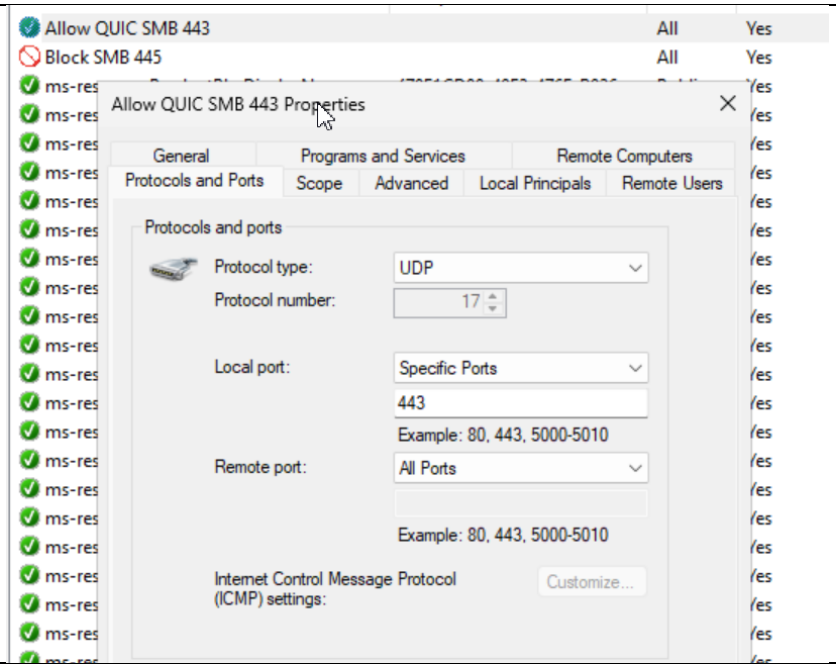
Recommended OS: Server 2025.

- Installed with default Roles, Workgroup in DMZ
- IIS Role is **not** required
- Inbound Firewall NAT: UDP, Port:443

Disk Layout: dedicated Data Disk for Quic repository recommended but C: drive can be used as well.



Firewall Rules

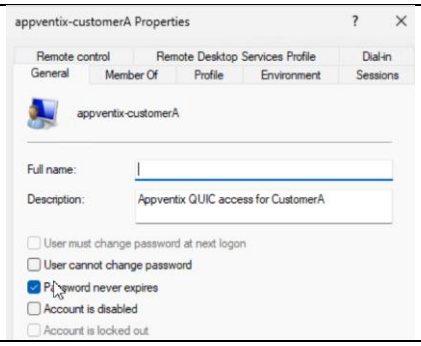
# Allow SMB port 443 UDP in Windows Firewall	New-NetFirewallRule -DisplayName "Allow QUIC SMB 443" -Direction Inbound -Protocol UDP -LocalPort 443 -Action Allow -Profile Any
# Verify the rule	Get-NetFirewallRule -DisplayName "Allow QUIC SMB 443" | Get-NetFirewallPortFilter
	

	PS C:\Users\Administrator> Get-NetFirewallRule -DisplayName "Allow QUIC SMB 443" | Get-NetFirewallPortFilter Protocol : UDP LocalPort : 443 RemotePort : Any IcmpType : Any DynamicTarget : Any

Optional: (if quic server is accessible / configured also from internal network, block Port 445 or SMB Clients may try to connect over this port instead of using the QUIC protocol).

# Block SMB port 445 in Windows Firewall	New-NetFirewallRule -DisplayName "Block SMB 445" -Direction Inbound -Protocol TCP -LocalPort 445 -Action Block -Profile Any
# Verify the rule	Get-NetFirewallRule -DisplayName "Block SMB 445" | Get-NetFirewallPortFilter

User Accounts

# Create local user ****Security Note:**** This local user is for AppventiX Agent → share authentication only.	$Password = ConvertTo-SecureString "ComplexP@ssw0rd!" -AsPlainText -Force New-LocalUser -Name "appventix-customerA" -Password $Password -Description "Appventix QUIC access for CustomerA" -PasswordNeverExpires -AccountNeverExpires Name Enabled Description ---- ------- ----------- appventix-customerA True Appventix QUIC access for CustomerA
	

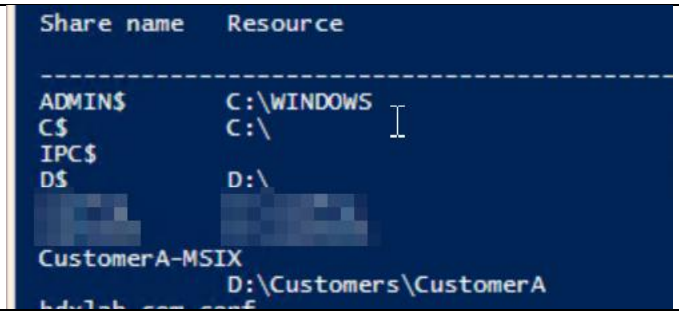
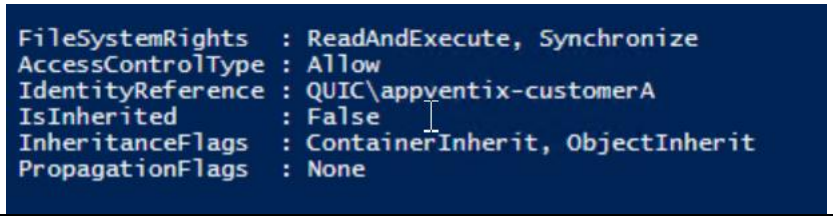
Create File Structure

# Create MSIX storage location	New-Item -Path "D:\Customers\CustomerA" -ItemType Directory New-SmbShare -Name "CustomerA-MSIX" -Path "D:\Customers\CustomerA" -FullAccess "Administrators" -ChangeAccess " appventix-customerA"
	Name ScopeName Path Description ---- --------- ---- ----------- CustomerA-MSIX * D:\Customers\CustomerA

NTFS Permissions

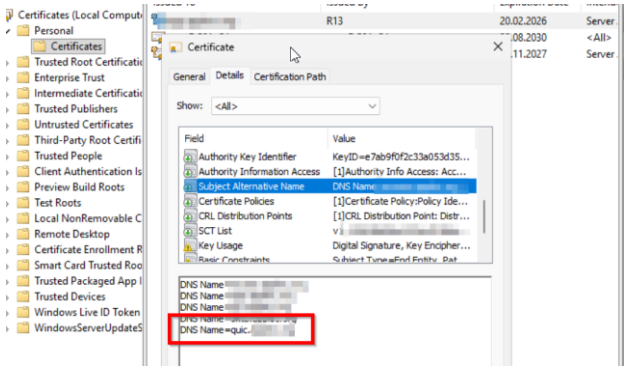
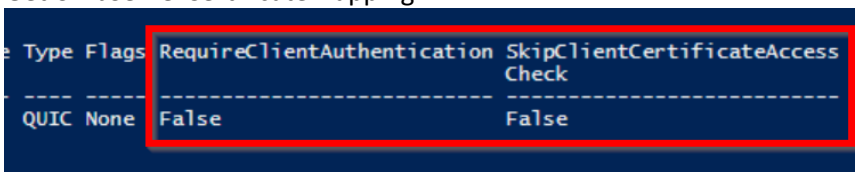
# Set NTFS permissions	
	$Path = "D:\Customers\CustomerA" $Acl = Get-Acl $Path $AccessRule = New-Object System.Security.AccessControl.FileSystemAccessRule("appventix-customerA", "ReadAndExecute", "ContainerInherit,ObjectInherit", "None", "Allow") $Acl.SetAccessRule($AccessRule) Set-Acl $Path $Acl

Validate the Share and Permissions

Net share	
Validate permissions	(Get-Acl $Path).Access | Where-Object IdentityReference -like "*customerA*" 

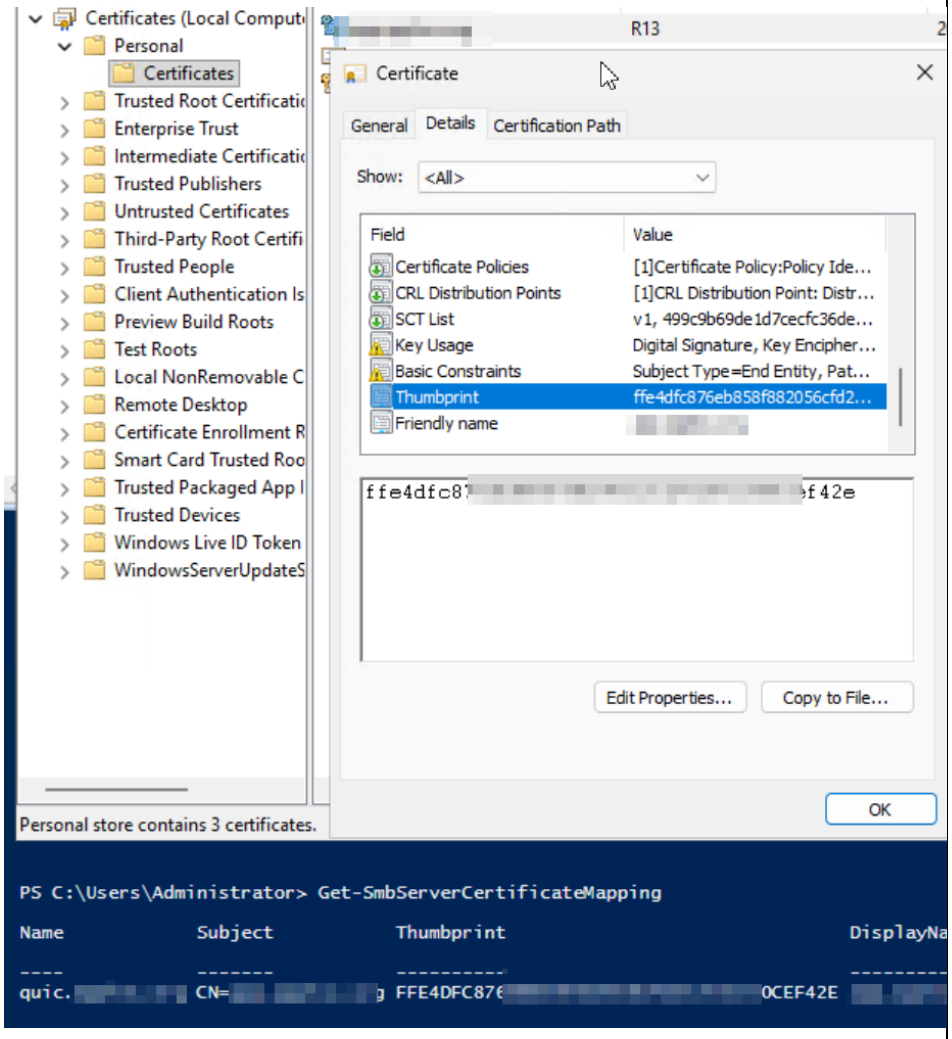
SSL Cert

For QUIC a normal server SSL certificate is enough, just like one which is used on a webserver. This can be a certificate from a third party SSL vendor like Sectigo, but also a certificate created by your internal PKI environment.

#import SSL Certificate for QUIC Listener	$cert=Get-ChildItem -Path "Cert:\LocalMachine\My" \| Where-Object Thumbprint -eq 0880DFBC3FB138CB9E598C87CCC3B53772021694 #$hash = $cert.GetCertHashString("SHA256") 
#map cert to namespace	New-SmbServerCertificateMapping -Name quic.mycorp.com -Thumbprint $cert.Thumbprint -StoreName My
Configure SMB Server Certificate mapping	Set-SmbServerCertificateMapping -Name quic.mycorp.com -RequireClientAuthentication $false Set-SmbServerCertificateMapping -Name quic.mycorp.com -SkipClientCertificateAccessCheck $false
Note: If the certificate contains multiple SANs this command might show the default Name here. Can be ignored	Get-SmbServerCertificateMapping  Name Subject Thumbprint DisplayName StoreName Type Flags RequireClientAuthentication SkipClientCertificateAccess Check ---- ------- ---------- ---------- --------- ---- ----- ---------- ---------- quic.mycorp.com CN=quic.mycorp.com FFE4DFC876EB858F882056CFD25255CE50CEF42E quic.mycorp.com My QUIC None False False

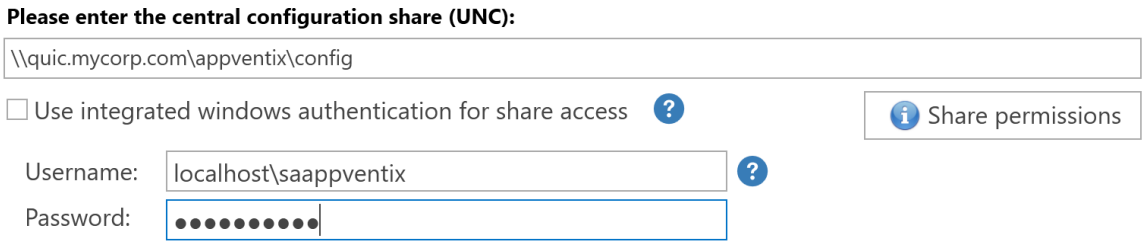
	Get-SmbClientAccessToServer -Name quic.mycorp.com

Check Config

Check Listener on UDP Port	netstat -na | Select-String "UDP" | Select-String ":443"
Listener processes	netstat -nao | Select-String "^ *UDP.*:443" (returns PID 4 for example) tasklist /FI "PID eq 4" Image Name PID Session Name Session# Mem Usage System 4 Services 0 164 K
Check SSL Cert mapping Thumbprint must match the SAN Cert in machine store	Get-SmbServerCertificateMapping 
Check config	Get-SmbServerConfiguration

Most important	EnableSMB1Protocol : False EnableSMB2Protocol : True EnableSMBQUIC : True EnableStrictNameChecking : True	

Configure the QUIC share in the Central View console:



Please contact support@appventix.com if you have any questions regarding the QUIC share implementation.

Advanced configurations

AppVentiX should work out of the box with default settings, but every environment has different requirements and characteristics. AppVentiX can be customized to adjust to those requirements.

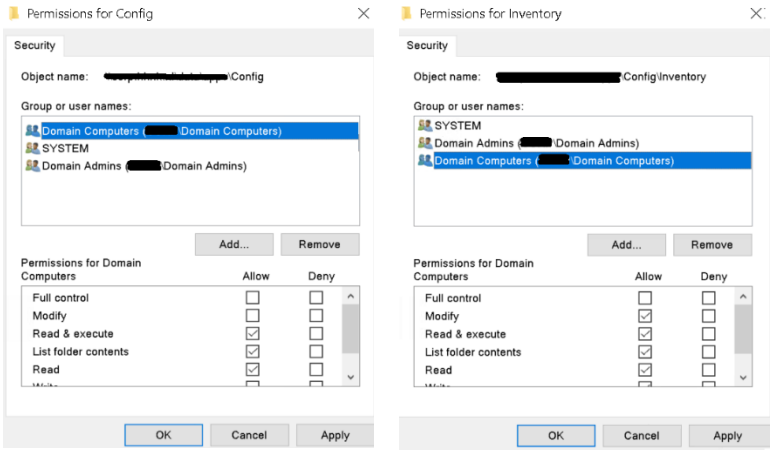
Share permissions and share configuration

AppVentiX supports **Windows file shares (both direct and DFS), Azure file shares (domain integrated or stand-alone) and shares created on storage vendors like Nutanix and NetApp**. AppVentiX can be configured to use integrated authentication or a (service) account to access the shares. The permission table is visible in the Central View console (share permission button). A screenshot is added at the beginning of this guide. The following share permissions are needed for AppVentiX to operate:

Share	AppVentiX Agent	AppVentiX Central View
Configuration share	Read	Read\Write*
Configuration share (inventory folder)	Read\Write	Read\Write
Content share(s)	Read	Read*

* Central View needs also write permissions if you want to convert MSIX packages to app attach or App-V packages to MSIX, if you want to delete content from the Content Share directly from the console you also need write permissions. For other management activities only read permissions are needed. This means you can provide the console to helpdesk\admins to operate the deployment but not change any configurations.

When creating a share in an active directory environment, you can use integrated authentication or provide a (service) account. When using integrated authentication, provide the domain computers group read permissions on the configuration share and read\write permissions on the inventory folder inside the configuration share. Below a screenshot of the permissions:

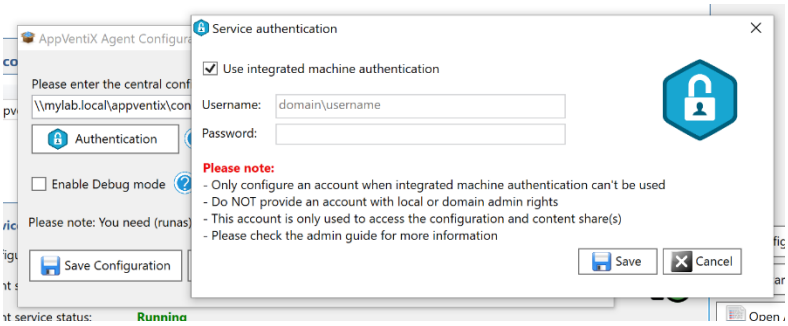


Domain computers: Read permissions on configuration share
Domain computers: Read\Write permissions on inventory folder
Domain admins (or group that performs management): Read\Write permissions

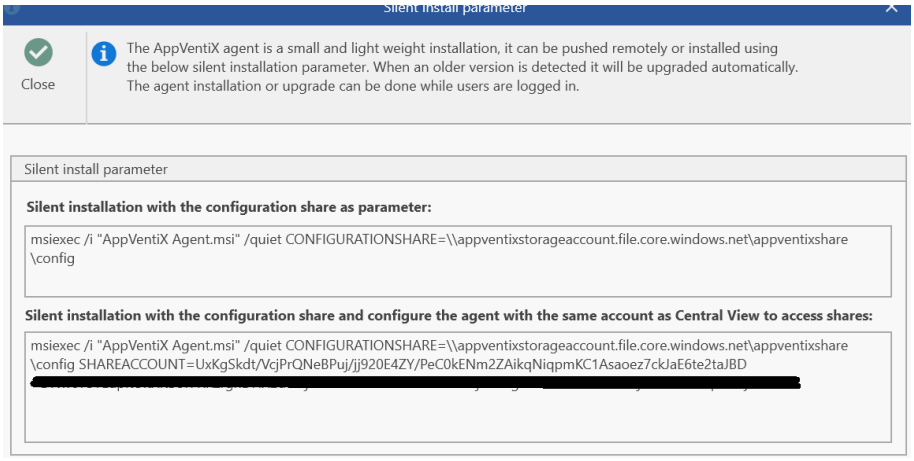
Share permissions can be set to everybody full control, so NTFS permissions are used for effective permissions to the files.

When the machines are Active Directory integrated, the AppVentiX agent uses the computer account to query AD to retrieve user groups. By default this works out of the box, but sometimes Active Directory is configured with a lot of security related settings. In this case contact support@appventix.com so we can help with the configuration of the AD domain integration.

The agent also uses integrated authenticated by default, integrated authentication uses the machine account to read the shares. You can configure the agent with a user account, just like in the Central View console:



You can also install\update the agent silently to use the same account as configured in the Central View console (click on the silent install button in the Central View console):



This makes it easy to deploy the solution with minimal effort and get up and running quickly.

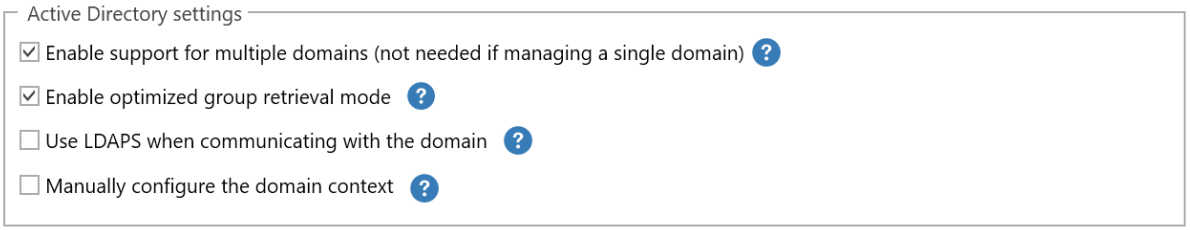
Central View inventory

By default the machine, user, process and event inventory data is stored on the configuration share in the inventory folder. When an inventory is triggered the agent will receive a command (also through the configuration share) to perform the inventory. The inventory data is displayed (together with the inventory time) in the console.

Optionally it's possible to enable direct inventory to the agent using WinRM (in the advanced agent settings turn the inventory slider to disabled). The performance and scalability of the inventory through the configuration share is much better then the direct connection, so recommended is to leave the inventory through configuration share option enabled.

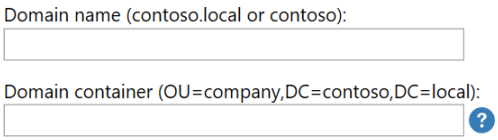
Central View advanced settings

In the Central View advanced settings you can configure global settings which will apply to the whole deployment (both console and agents), the following advanced settings can be configured:



- Multi domain support: This setting is enabled by default for new installations. It will provide support when users are located in different domains. It's not needed if you manage a single domain.
- Enable optimized group retrieval will optimize the group retrieval for users, using a token refresh mechanism to also update groups when users are logged in

- Enable LDAPS: When enabled, both Central View and agents will use LDAPS to communicate to the domain, please note you need certificates installed on your domain controllers
- Use SSL for remote machine inventory: This setting will enable WinRM communication over SSL. Please note you have to deploy a machine certificate to the agent machines in order to use WinRM over SSL. On the next page you will find more information how to enable WinRM over SSL. This setting is not used when inventory is done through configuration share (new default)
- Inventory machines on FQDN, when enabled the FQDN (for example machinename.mydomain.local) name will be used to contact the machines for management
- Remote connection (WinRM) time-out value this is the maximum time to wait before timing out a connection to an agent, increase this value when running inventory over WAN connections. This setting is not used when inventory is done through configuration share (new default)
- Manually configure the domain context: By default AppVentiX will use the default domain configuration, but when your Active Directory is complex or when it has delegation configured you might want to manually configure the domain context. When you enable this option you can configure the domain name and domain container:



For example the domain name you want to use is: contoso
And the domain container is OU=Myorganization,DC=contoso,DC=local
AppVentiX will look for machines and user\groups in this OU only, it's also possible to provide multiple domain containers, for example when your machines and user groups are in different OU's, you can configure the domain context as follows:

OU=Mymachines,OU=Myorganization,DC=contoso,DC=local;OU=Myusers, OU=Myorganization,DC=contoso,DC=local

(note the separating semicolon)

Please contact support@appventix.com if you need any help with this configuration.

- Inventory machines on FQDN instead of netbios name, enable this setting when your machines cannot be reached on machine name only, when enabled Central View will retrieve the FQDN name of the machine from AD and will show this in the inventory instead of only the machine name, it will use the FQDN when contacting the machine

Supported operating systems

The following operating systems are supported by the AppVentiX agent:

- Windows server 2016 (64Bit)
- Windows server 2019 (64Bit) (App-V and MSIX client are embedded in Server 2019)
- Windows server 2022 (64Bit) (App-V and MSIX client are embedded in Server 2022)
- Windows server 2025 (64Bit) (App-V and MSIX client are embedded in Server 2022)

- Windows 7, 10 and 11 (64Bit) (App-V and MSIX client are embedded in Win10\11)

The following operating systems are supported by the AppVentiX Central View console:

- Windows server 2019 (64Bit)
- Windows server 2022 (64Bit)
- Windows server 2025 (64Bit)
- Windows 10 (64Bit)
- Windows 11 (64Bit)

For all AppVentiX components you need at least .NET 4.8.

Please check the MSIX and MSIX app attach compatibility with the OS you are using, app attach is currently only supported in Windows 10 Build 2004 and up. And MSIX shared containers are only supported on Windows 10 Build 21X and up.

Upgrade from earlier version of AppVentiX

The upgrade from an earlier AppVentiX version to the latest version is straight forward, the components are upgraded in place (no need to uninstall first). Before upgrading create a copy\backup of the configuration share.

Upgrade the Central View console first, then the agents. The version of the agent is always 1 major version backward compatible with the Central View console, this means you don't have to upgrade all agents at once, but it's recommended to keep this period as short as possible.

FSLogix and roaming profile settings

AppVentiX has been verified to work very well with FSlogix and other profile management solutions like Citrix UPM and Windows roaming profiles, contact support@appventix.com if you have any questions. If you encounter any issues, please check the following:

FSlogix in combination with App-V

When using FSLogix in combination with App-V you might encounter some packages that do not work anymore after the second logon. To fix this, enable the "Always publish this package" option in the publishing task. This will make sure the package integrations for the affected package is configured correctly after each login. Another work around is to add the following path to the FSlogix exclusions xml file: AppData\Local\Microsoft\AppV\Client.
Also make sure to use the latest FSLogix build.

FSlogix in combination with MSIX

MSIX data roaming is enabled by default for FSlogix (see below screenshot). No additional configuration is needed. MSIX data is roamed for normal deployed MSIX packages and MSIX packages delivered by app attach.



The log file for the roam MSIX app data can be found in: %appdata%\AppVentiX
The log file will be cleared every 7 days.

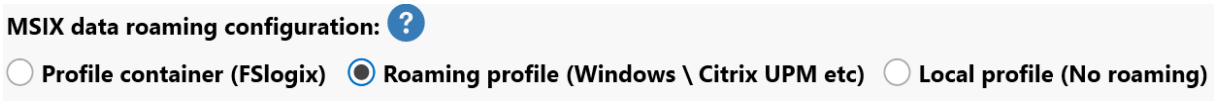
There is only one exclusion needed when you encounter stale Windows Startmenu shortcuts, then create or modify your existing FSlogix exclusions xml file with the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<FrxProfileFolderRedirection ExcludeCommonFolders="0">
<Excludes>
<Exclude
Copy="0">AppData\Local\Packages\Microsoft.Windows.StartMenuExperienceHost_cw5n1h2txyewy\TempState</Exclude>
</Excludes>
</FrxProfileFolderRedirection>
```

Please note: Above exclusion is now default in FSlogix since 2501 release.

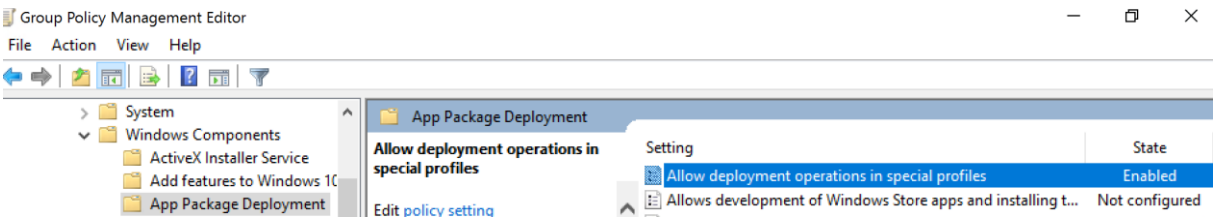
Windows roaming profiles with MSIX

For Windows roaming profiles the following setting needs to be enabled:



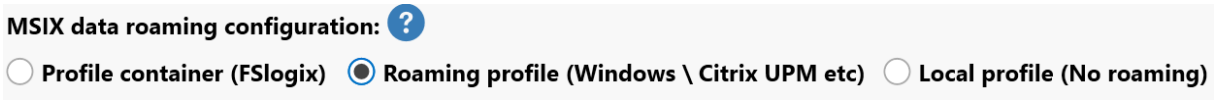
When you configured the roaming profile settings to delete cached copies of the profile, you need to enable the below GPO setting as well:

Allow deployment operations in special profiles



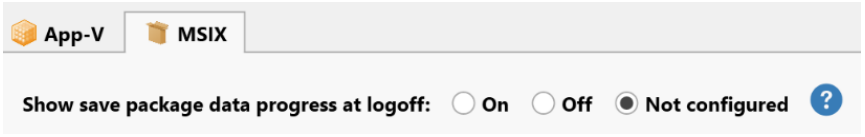
Other profile solutions that capture the roaming data of the user (Like Citrix UPM) with MSIX

Enable the following agent setting:



Make sure the %appdata%\AppVentiX folder is captured\roamed in the roaming profile solution.

Optionally the show save package data progress at logoff can be enabled. This will show a brief notification of the MSIX data being saved at logoff.



Automated image building and deployment actions

Image build events

AppVentiX is often used as part of automated image building procedures. To check if AppVentiX has finished pre-caching packages in the image, the below PowerShell script can be used. The script will check the AppVentiX finished event and then continue.

For App-V, the check can be performed on the pre-cached all App-V packages event (122):

```
$eventlogname = "AppVentiX Agent"
# Check for all packages pre-cached event
Do {
$allpackagesloadedevent = Get-EventLog -LogName $eventlogname -Source "APPV" -Newest 5 |
where {$_.InstanceId -eq 122}
if($allpackagesloadedevent)
{Write-Host "Finished, all App-V packages are pre-cached"}
else{
Write-Host "Not finished, sleep 30 seconds"
Start-Sleep -Seconds 30
}
} # End of 'Do'
While (!$allpackagesloadedevent)
```

For MSIX, the check can be performed on the refresh cycle finished event (2000):

```
$eventlogname = "AppVentiX Agent"
# Check for refresh cycle finished event
Do {
$allpackagesloadedevent = Get-EventLog -LogName $eventlogname -Source " Service" -Newest 5 |
where {$_.InstanceId -eq 2000}
if($allpackagesloadedevent)
{Write-Host "Refresh cycle finished, all MSIX packages are pre-cached"}
else{
Write-Host "Not finished, sleep 30 seconds"
Start-Sleep -Seconds 30
}
} # End of 'Do'
While (!$allpackagesloadedevent)
```

Note for MSIX app attach: It's not recommended to place app attach packages on a content share configured for a machine group where the build VM is member of. App attach packages are not needed to pre-cache in the image because they are virtual disks attached to the VM in runtime.

Run the refresh cycle from the command line

The refresh cycle automatically runs when the machine boots, the refresh cycle can also be invoked from the Agent GUI, based on a timer or remotely through the Central View console (recommended). It's also possible to invoke the refresh cycle with PowerShell. This can be done using the following PowerShell one liner:

```
(Get-Service 'AppVentiXService').ExecuteCommand(254)
```

When Central View console takes longer to start

When the machine running the Central View console doesn't have internet access, the console can be slow to start. This is default behaviour for .NET applications that are signed with a certificate, because there is no internet connection the Microsoft CRL checking process can't check the certificate used to sign the AppVentiX executables. To work around the issue you can manually update the CRL, enable internet access or disable CRL checking:

```
Disable CRL Checking Machine-Wide Control Panel -> Internet Options -> Advanced -> Under security, uncheck the Check for publisher's certificate revocation option
```

Please note that AppVentiX doesn't make any internet (outbound) connections at all, so internet access is not needed for AppVentiX to work.

Also please make sure to check the connection to the central configuration share. How further away the Central View console is placed from the central configuration share it can take longer to retrieve the configuration.

Example configurations of the AppVentiX Agent

The following examples can give you some insight into how AppVentiX can be configured. This configurations are validated to work, but they are only intended to give you an idea of the possibilities. They can be used as guide line but are not written down here to serve as best practice or recommended configuration. Feel free to contact support@appventix.com we are always happy to discuss which approach is the best for your environment.

Example of AppVentiX deployment in combination with non-persistent machines and App-V

- Move the Cache to a persistent drive (for example the same one as the write-cache)
- Configure in the AppVentiX agent settings to detect the image state (don't deploy packages when in read\write mode)
- Configure the agent to clean the cache after reboot (needed because drive is persistent)
- Use SCS mode in combination with the mount specific packages option (mount packages that either perform better when fully cached or if you want them to be higher available) this combination gives you the best of both worlds

Example of AppVentiX deployment in large scale non-persistent VDI environment in combination with App-V

- Keep the cache on the default location
- Enable the detect image state option and configure to mount all packages in the cache in private mode and use SCS mode when in read-only mode
- The rest of the configuration will be configured automatically according to above setting

In this deployment mode, when the image is booted in private mode (during Windows update or build update), all the latest packages will be mounted in the cache automatically. An event will be logged when the pre-cache operation is done (you can configure your automatic build method to check for this event, contact support for an example).

When the image is in read-only mode new packages can be deployed to the machines, in read-only mode new packages will be added automatically by using SCS mode (reading package content directly from the share). In this way the write-cache isn’t polluted.

This deployment mode gives you the best of both worlds in large VDI deployments: lower IO during boot and be able to deploy packages and application updates during runtime of the machine.

Example of AppVentiX deployment in combination with persistent machines and App-V (like RDS\AVD and physical machines)

- Keep the cache on the default location
- Don't clean the cache at machine reboot
- Configure the AppVentiX agent to remove packages that are no longer on source share (keep cache in balance with source)
- Use SCS mode in combination with the mount specific packages option (mount packages that either perform better when fully cached or if you want them to be higher available) this combination gives you the best of both worlds. Or mount all packages by default if disk space is not an issue

Example of AppVentiX deployment in combination with non-persistent machines and MSIX

- Leave the MSIX cache location default or redirect to persistent drive in the agent settings
- Configure the AppVentiX agent to detect the image state (agent setting) (don't deploy packages when in read\write mode). In this scenario we only deploy packages when the image is readonly.
- You can use both MSIX app attach and normal MSIX package delivery, normal packages will be cached on the persistent drive or temporary write cache location (depending if you redirected the cache in the agent settings)
- Clean the MSIX cache after reboot (agent setting) when redirected to persistent drive
- Using above combination gives you the best of both worlds in terms of space utilization and performance

Example of AppVentiX deployment in large scale non-persistent VDI environment in combination with MSIX

- Keep the cache on the default location
- Enable the detect image state option and configure to load all packages in the cache in private mode
- Place all MSIX packages on a content share with the pre-cache option enabled
- Place MSIX app attach packages on a content share with pre-cache disabled

In this deployment mode, when the image is booted in private mode (during Windows update or build update), all the latest MSIX packages will be loaded in the cache automatically. An event will be logged when the pre-cache operation is done (you can configure your automatic build method to check for this event, contact support for an example).

When the image is in read-only mode you can delivery MSIX packages through app attach to users, this will not consume any disk space. You can also decide to deliver all packages through app attach (like previous example) but this example will give you an indication about the possibilities with AppVentiX to achieve the best possible combination.

This deployment mode gives you the best of both worlds in large VDI deployments: lower IO during boot (MSIX packages already exists in image) and you will be able to deploy packages and apply application updates during runtime of the machine.

Example of AppVentiX deployment in combination with persistent machines and MSIX (like RDS\AVD and physical machines)

- Keep the cache on the default location
- Don't clean the cache at machine reboot
- Configure the AppVentiX agent to remove packages that are no longer on source share (keep cache in balance with source). This is enabled by default.
- You can use a combination of normal MSIX deployment and MSIX app attach:
- Use normal MSIX delivery to have less pressure on the content share (packages are loaded in the cache and published from there to users). Also when the share goes down there is less downtime (app attach needs constant availability to the content share). Because packages are published from the local cache of the machine, the application might work faster.
- Use app attach to save disk space on the machines and speed up the registration of bigger packages (the package doesn’t have to be loaded on the machine, package data will be accessed over the network through the attached disk

With AppVentiX you are in complete control over the deployment there is always a combination available that suites your uses case the best. Feel free to contact support@appventix.com we are always happy to discuss which approach is the best for your environment.

Website : www.appventix.com
Sales : sales@appventix.com
Support : support@appventix.com