Chocolatey For Automate 3.6

Chocolatey for Automate is a software package manager for Windows that allows the MSP or Management team to deploy and update common software titles across the RMM without the need for daily management of the tool.

Once a client is enabled and configured, Chocolatey for automate will automate the deployment of Chocolatey across the environment, manage all repository sources configure, auto install approved software if configured to do so and update any packages managed by Chocolatey whether natively installed or managed through Chocolatey for Automate.

You can use the plugin to manually install any package available and to process updates allowing the engineer quick access to deploying needed tools and updates to the agent without disturbing the customer and user of the PC.

New in Chocolatey for Automate 3.6

One of the new items in Chocolatey for Automate 3.6 is a redesign of the automation services. In previous versions the Automation was tied into Automation Sync Services which at times could load down Automate as it tries processing thousands of script schedule requests. In Chocolatey for Automate 3.6 most of the agent automation has been moved over to Automate Groups and Searches. Plugins4Automate provides Automate with a selection of Groups that use Auto Join Searches to collect and manage framework deployment, approved application installs, Source and UNC cache management. As agents join these groups during automation cycles, they groups will assign a smaller script to manage that function.

Schedule daily, weekly or monthly updates to best fit your maintenance schedules. Plugins4Automate has updated the provided “Approved packages” list to include the most common Chocolatey extensions. Preinstalling these extensions allows the agent to have the needed dependencies that packages like Google Chrome will require to complete installs and or updates.

Upgrading from Chocolatey for Automate 3.5 and older

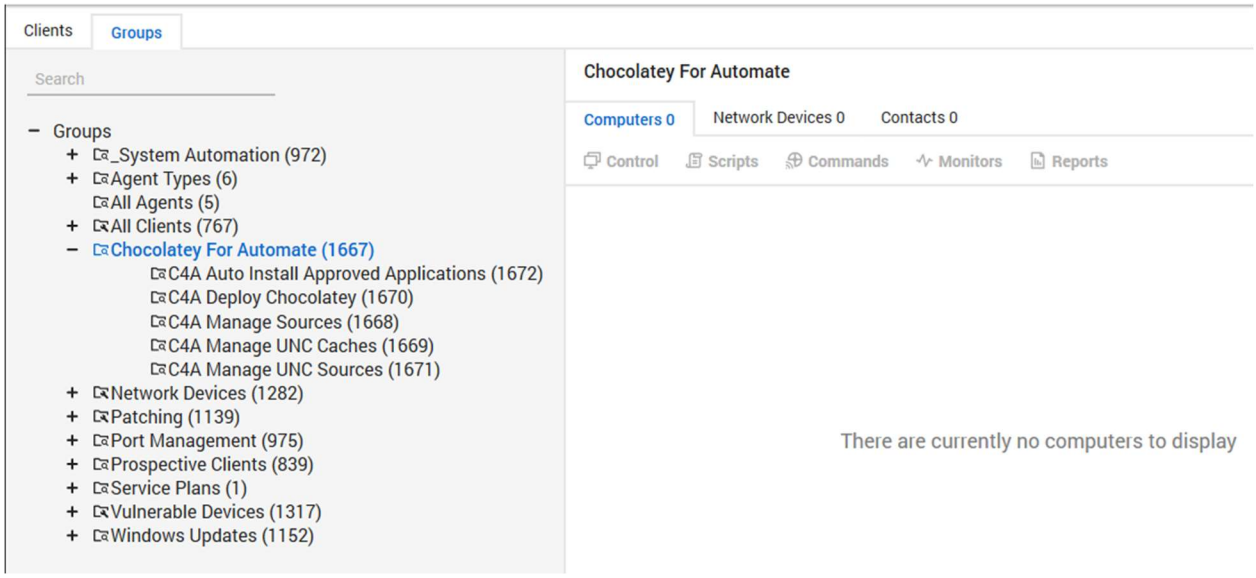
If you are upgrading from a previous version of Chocolatey for Automate then you will need to do a little house cleaning in order to get the plugin to be fully functional.

You will need to drop the following tables in MySQL before performing plugin update.

- plugin_p4a_chocolatey_applications
- plugin_p4a_chocolatey_cache
- plugin_p4a_chocolatey_clients
- plugin_p4a_chocolatey_enable
- plugin_p4a_chocolatey_properties
- plugin_p4a_chocolatey_repos

Getting to know your groups.

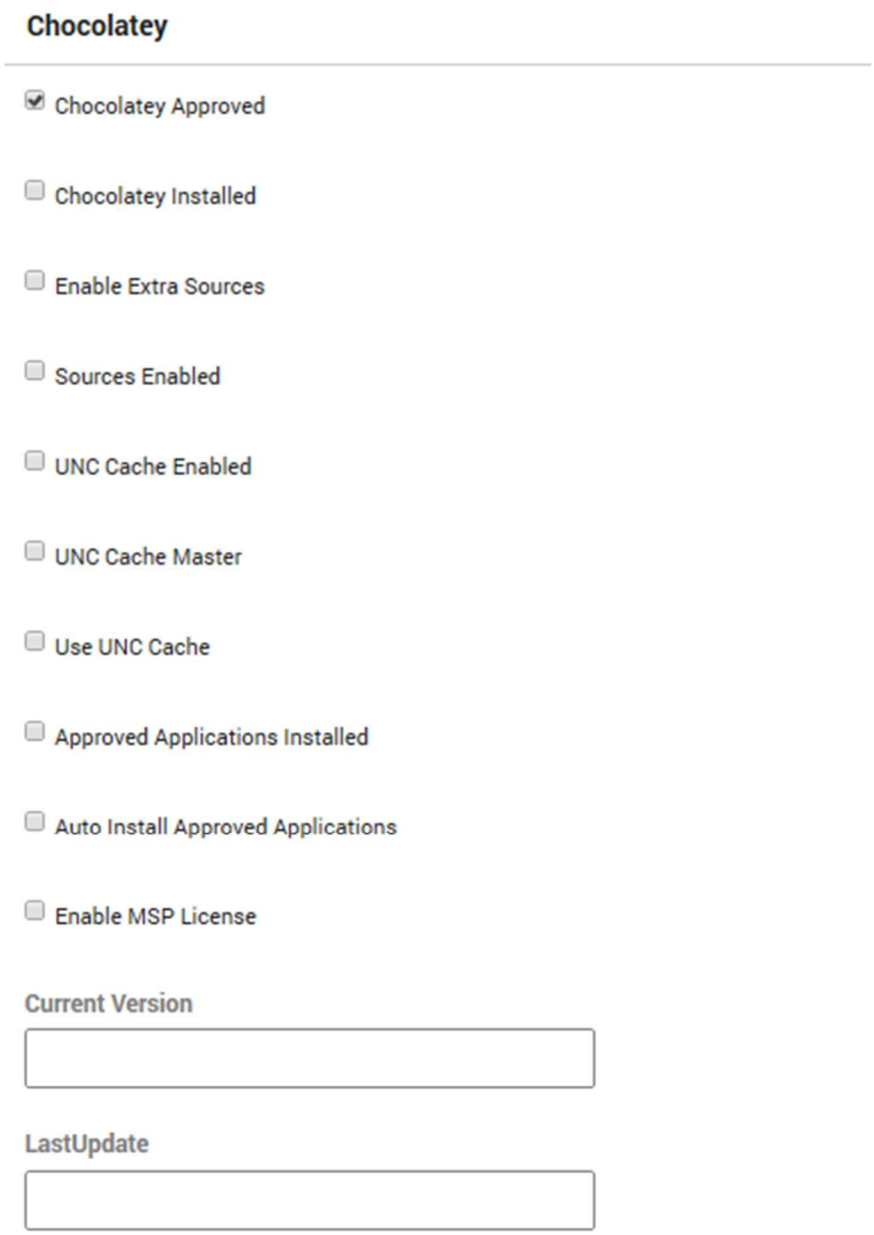
Under your Groups tab in Automate, you will find the **Chocolatey for Automate** group. This top-level group manages the daily packages validations collected from each agent. Under this group are several subgroups. Each of these subgroups manages a function. Agents will come and go from groups as they navigate through all the different processes needed to provide the agent with updates on a regular basis.



Get to know your EDFs

Each agent is given a group of EDFs that sets different states. These states are controlled with the plugin and do not require direct management but if you need to force a state change for any reason you can do that here.

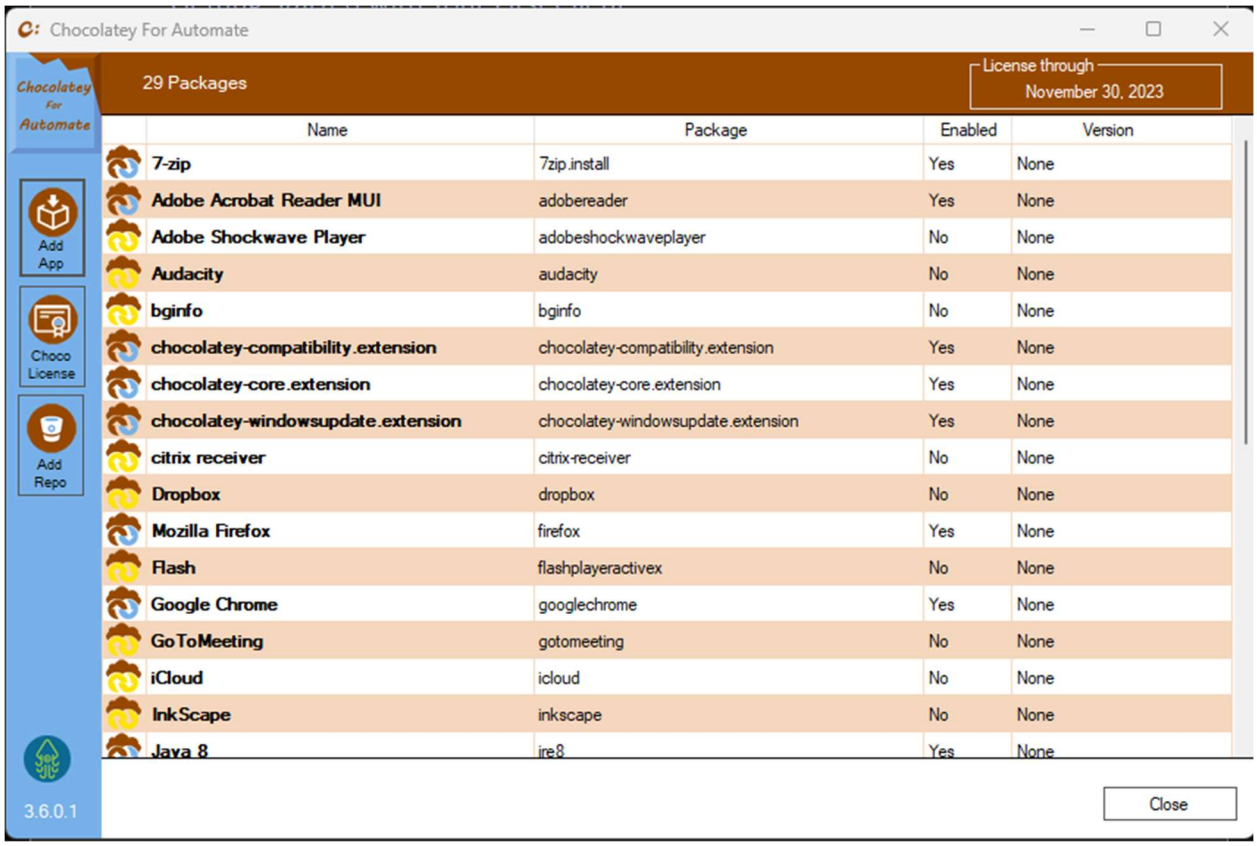
An example of a state change would be to uncheck **Chocolatey Installed**. This will cause the agent to auto join the **C4A Deploy Chocolatey** subgroup which would test for Chocolatey and install (if needed). Once installed or verified installed this EDF will get rechecked automatically causing agent to leave the **C4A Deploy Chocolatey** subgroup.



Getting Started with Your First Client.

To get started with Chocolatey for Automate, launch the Control Center Console and from the main Tools menu, select the Chocolatey for Automate Tool. We have pre-populated the Approved Packages list with a bunch of the more popular software titles. You can remove any applications you do not need or leave them disabled. Leaving packages disabled is preferred as it will help populate repository version numbers for agents that may have that package already installed. If the client has been deployed and agents are scanned for packages, you may find agents missing repo versions of those packages. The agent will continue to report the installed version, but repo version may be blank. To resolve this, add that package to the approved packages list and set it as disabled if you do not want it distributed to agents by default.

You will also see your license for the plugin displayed in the top right side of this window.



Name	Package	Enabled	Version
7-zip	7zip.install	Yes	None
Adobe Acrobat Reader MUI	adobereader	Yes	None
Adobe Shockwave Player	adobeshockwaveplayer	No	None
Audacity	audacity	No	None
bginfo	bginfo	No	None
chocolatey-compatibility.extension	chocolatey-compatibility.extension	Yes	None
chocolatey-core.extension	chocolatey-core.extension	Yes	None
chocolatey-windowsupdate.extension	chocolatey-windowsupdate.extension	Yes	None
citrix receiver	citrix-receiver	No	None
Dropbox	dropbox	No	None
Mozilla Firefox	firefox	Yes	None
Flash	flashplayeractivex	No	None
Google Chrome	googlechrome	Yes	None
GoToMeeting	gotomeeting	No	None
iCloud	icloud	No	None
InkScape	inkscape	No	None
Java 8	jre8	Yes	None

Managing Sources

By default, all Chocolatey agents will use the public repository. You can choose to also include other repositories like your own caching repository or simple UNC shares. I will talk more about the different repositories further in this document. For now, selecting the “Add Repo” button will allow you to place repositories that use the web (http(s)) as an address for package service.

Manage Chocolatey MSP License

This is an optional service and is not required to operate and use Chocolatey with this plugin. The MSP edition of Chocolatey is an added license file that is placed in a directory under the Chocolatey framework that unlocks added features of Chocolatey. Our plugin gives you the option to be able to save that XML file to Automate and to deploy it to agents as you desire.

Configure your Client

Once you have enabled and disabled the application list to your liking, you can navigate over to your first client to enable for Chocolatey. In the Control Center Console double click any client from your client list to launch the client console for that client. Look for the Chocolatey for Automate tab and select it.

Next select to enable Workstations and wait for agent list to build. Once your agent list is visible select the agents you want to disable any use of Chocolatey on if you have any. By default, all agents will load up as enabled for Chocolatey.

If you stopped here and did nothing more, all agents selected would install Chocolatey and do a local package scan. If the agent already had Chocolatey installed, then the version would be noted an agent enable. Once enabled all agents will scan for existing software that's been installed by Chocolatey. No software would be pushed from the approved software list, the agents would only scan for manually installed or pre-existing software titles. This information would be displayed in the panel to the right once scans have been completed. Scans are handled by the master group Chocolatey for Automate.

Whether or not you are using UNC Caching for any client you should have at least one Cache Master set across the RMM. It can be set on any client and any agent that is normally online. If you are using UNC caching, then you should already have a Cache master for that location. If so, then you will not need to further set one. To set an agent as a Caching Master select the agent from the list and right click to pop up menu. From the menu select **Set Agent as Cache Master.**

With this being your first client to enable you maybe wanting to take things slow until you get the hang of what the plugin is doing and how. In this case you might allow this client 24 hours to filter in and out of the groups to get deployed and tested followed by a scan for packages. This will allow the agents time to process everything and return data if it exists.

Installing Approved Software

To have agents install approved software, select the "Auto Install Approved Packages" check box. This will cause each agent to join the **C4A Auto Install Approved Applications** subgroup. This group will cause the (enabled) Approved Packages list to be passed down to the agent and installed based on the agents

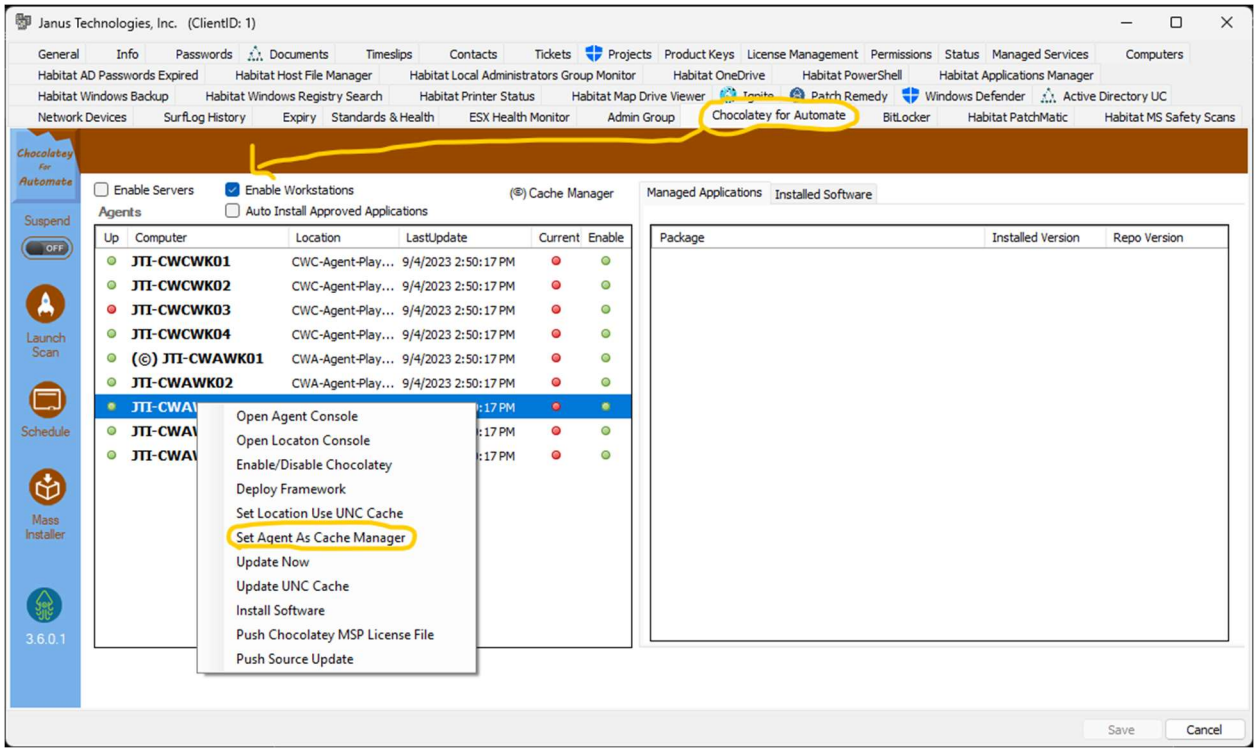
currently set sources. If no sources or UNC cache shares were configured for the agent, then it will use the default public repository. This is a group script function and runs daily, allowing a little time for group actions to automate. Once packages are installed the agent will auto leave this group. It is normal for this group to be empty.

If you are using Source Repos, they are a global setting that requires you to set each client to use Active Sources. At the Client console select from agent menu to Push Source Updates. This will cause the agents to auto join the subgroup **C4A Manage Sources,** this group deploys any active sources to the agent's Chocolatey configurations. The sources will be added with the priority provided. Based on priority the default repository may still be used for packages not found on source. Once sources are deployed agent will leave this group. It is normal for this group to be empty even though you are using sources.

Schedule Updates

Updates default to daily checks but if you would prefer to some control over when updates are fired off then you can select the Schedule button in the left-hand menu of client console. You can schedule daily, weekly (pick the day) or monthly (pick the day) that updates are performed on agents under that client.

There are several manual controls that will allow you to force actions on agents including installing packages in mass, installing packages to a single agent, updating agent, updating UNC cache at a location and many others.



Avoid Rate Limiting from Public Repo

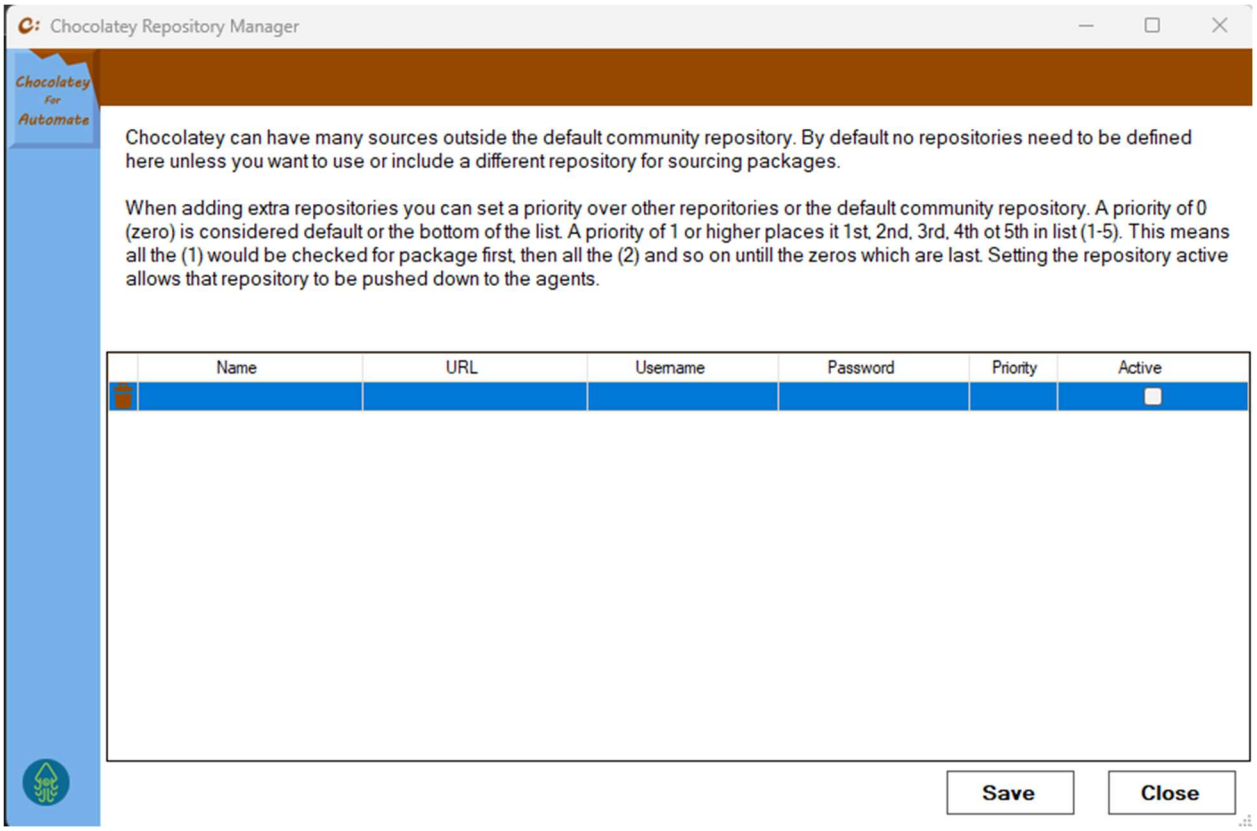
By default, the plugin uses the Chocolatey Framework as it is installed. This means that any packages installed or updated will originate from the default community repository. For large locations where many agents come out from 1 network address this could cause you some issues actually getting agents software deployed or updated.

Avoid Rate Limiting from Chocolatey.org

Chocolatey.org has strict rate-limiting policies, which means you may start receiving that dreaded "429 error" in the middle of deploying or upgrade packages from agents at locations that have reached these limits.

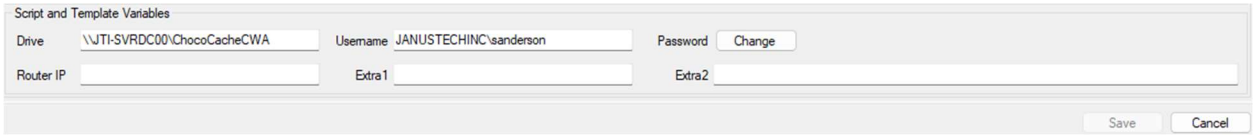
We have 2 ways to avoid the 429 issue

1. **Run a $7 caching server for your entire MSP (all clients)**
   Take a look at our "Create your own Chocolatey Caching Host" document included in plugin zip package. This makes caching quick and easy, add the new source to the plugin and let the client console know to use sources. We will take care of the rest by deploying that source out to each agent. When an agent is told to install or update a package, it will first go to the cache server to get package manifest.



2. **UNC Cache Share**
   This solution is managed by client location, create a simple UNC share on the local network that you can point agents to retrieve Chocolatey package manifests. Set a caching master for each location to handle making sure the cache is kept up to date. Then only the caching agent will communicate with the public repo preventing the dreaded 429 errors. Manage the UNC path and permissions under the Drive variables provided in the location consoles.

   Once configured the plugin automation will manage the cache making sure its populated and updated regularly.



Troubleshooting

As much as we would like it, there is always room for an issue to arise. Use the EDFs under the agent console to verify that current state of any agent. When sending in support requests make sure to take a screenshot of the agent's EDFs for the support team to review. Also collect any agent script and command logs and provide them with the support request. Also document what groups that agent is currently joined to.