Scalability and Performance Considerations

Quest Access Manager is designed as a heavily distributed application, leveraging its deployment of agents to provide decentralized storage and processing. Each agent maintains the information it collects on its hosting computer. Listed below are a few key criteria to deploy Access Manager in a given environment.

Agents should be close to their targets

Agents perform best when they're as close as possible to the resources they're indexing. Local agents executing directly on file servers provide the best performance, because there is no network stack involved. In the case of NAS devices, or file servers where it is not practical to install agents, ensure that agents are installed on as low latency a connection to the target managed host as possible

How much can each Agent manage?

Each agent can generally be configured to manage between 50 and 100 million files and folders. The size of the file systems being indexed are generally not relevant; 5 million 1 Gb files are the same to the agent as 5 million 1 byte files; we don't read the content, only the metadata. Generally, agents will index their target file systems between 600 and 5000 items\second. This indexing performance (which can be seen through the agent metrics view, and the agents view in 2.1) can be used to roughly calculate how long a full indexing operation of a file system will take (dividing the total number of objects on the agent's roots by its indexing speed gives a rough time to completion.)

How many agents are required to monitor a managed host?

In a vast majority of cases, a single agent can effectively manage a standard file server. Only in cases where large scale NAS devices are used should multiple agents be leveraged (when the number of file system items is significantly greater than 50 million on the target.) Our testing has shown that unless the device is significantly powerful, increasing the number of agents targeting a device beyond two will offer only small increases in overall indexing performance. Going from one to two agents, however, can in some cases as much as double indexer performance.

Are there any restrictions in deployment for remote agents?

One of the strengths of remote agents is that multiple agents can be used to simultaneously draw information from a single server. In a number of situations, an agent will be given only so many resources on their targets. By leveraging multiple agents, more resources can be used by QAMs agents to increase their performance far beyond what could be attained by increasing thread counts for a single agent. As a consequence of this, more than one agent on a single hosting box cannot be used to index the same managed host. In cases where more than one agent is required to index a single device, multiple agent hosting systems must be used. However, given that the only resource significantly used by remote agents is network bandwidth, virtually any server can host one (including file servers hosting other local agents.)

One significant restriction of remote agents (not including those targeting NetApp and EMC Celerra devices) is that they do not support resource activity tracking. Because there are no services provided on windows servers to remotely detect activity as it occurs on the file system that includes identity information, this functionality is not supported in these scenarios in QAM.

What is the performance overhead and resource impact on the host machine of QAM Agents?

Resource usage during full file system indexing operations

When the QAM agent performs a full file system indexing operation, it reads the security descriptors of all files and folders using a number of threads. This operation can have an impact both on the CPU and IO subsystems of the agent host (as windows doesn't make reading security a terribly efficient operation.) It is not uncommon to see a QAM agent spike the CPU usage of it's hosting system by more than 50% during full file system scans. However, this by itself is generally not a cause for concern. The QAM agent sets it's process priority to a value much lower than other processes on the system, and will gracefully yield CPU to any other process requiring it. IO usage, on the other hand, can cause some more noticeable impacts. Because reading security from an NTFS volume involves a large amount of disk seeking, some file servers with weaker IO subsystems can encounter significant impacts to their ability to service disk requests. When this occurs, it is recommended that the scanningThreads property (mentioned in the Configuration Tweaks section) be used to lower the number of threads, thus decreasing the amount of seeks being performed by the disk subsystem.

Storage space used by agents

Generally, agents will use 100Mb\million files and folders stored. Note that the number of items encountered during an indexing operation is not necessarily the same as the number of items stored; to conserve space and ensure performance, the QAM agent avoids storing security information for any files which aren't considered "interesting" (that is, they only have inherited permissions.) The Agent Metrics view can provide insight into the amount of storage space being used by each agent, as well as an indication of the number of resources processed and stored. As of version 2.1, this information is also available through the Agents view.

Network usage

The following table describes the various operations performed between agents and the QAM server, and the approx. bandwidth and frequency of each.

Operation	Frequency	Approximate Bandwidth
Initial agent connection	Once per connected session (connections are re-established if severed.)	~3 Kb
Agent lease renewal	Every 4 minutes	~3 Kb
Initial security index synchronization	Once per connected session	1Kb – 30 Kb (depending on the number of trustees found on the target host)
Delta security index updates	Whenever required	1 Kb – 5Kb (depending on the number of pertinent changes.)
Resource activity recording	Once per granularity timespan (except when explicitly scheduled)	2 Kb – 5 Mb (depending on the amount of activity information recorded during the activity granularity window.)

Trustee Access Query	Whenever executed	5 Kb – 500 Mb (depending on the amount of information required to satisfy the query)
Resource Access Query	Whenever executed	5 Kb – 500 Mb (depending on the amount of information required to satisfy the query)

Remote Agents and full file system indexing operations

Remote agents, by their nature, need to pull security information from their target systems across the network to them locally in order to process it. Because of this, remote agents use significantly more bandwidth than their local counterparts. As a rule, it can be assumed that approx. 3Kb \ file and folder will be used to perform the scan. Because of this, it is strongly recommended that remote agents be as physically close to their target managed hosts as possible, to ensure ample amounts of bandwidth are available to them. Deploying remote agents where their targets are across WAN links, or even where their targets are in different data centers, is not recommended.

Remote activity tracking for NetApp and EMC devices

Because neither of these devices allow for the hosting of 3rd party software directly on the devices themselves, it is necessary to utilize their remote notification frameworks (FPolicy for NetApp devices, and CEPA\CAVA for EMC) to provide real-time security index updates and file system activity tracking. Because of the remote nature of these technologies, their use will impose a bandwidth overhead. However, the exact amount of this overhead varies significantly based upon the usage profile of the device. We have not identified any situations where this causes degradation of the performance of the target devices.

Activity tracking for local agents

In our testing, we have encountered overheads of less than 2% CPU usage when resource activity tracking has been enabled. Enabling this feature, however, does cause the agent to use more disk space on it's host. This is due to the temporary storage mechanism the agent uses when recording activity, in which the agent queues all activities it records in local databases, prior to aggregating the information and sending it to the server for long term storage. It is difficult to predict how much space will be consumed by these temporary storage files, but generally the amount does not exceed a few hundred MB at most. Note that prolonged disconnection of agents from the server when the agent has resource activity tracking enabled can cause these files to grow in an unbounded fashion.

Configuration Tweaks for Debug and Customization

The client Debug Interface

Some functionality, generally geared toward support and diagnostics, is not enabled in the UI by default. These functions, some of which can be fairly dangerous to use, can be enabled by creating a REG_SZ registry value called "debugInterfaceEnabled" at the following location in the registry:

HKEY_LOCAL_MACHINE\Software\Quest Software\Broadway\Client

No special data needs to be set in the value; it just has to exist. Once created, the QAM MMC client must be restarted for it to take effect. When enabled, the following functionality becomes available in the console:

- The “Client Identity Information” menu is added on the console root node. This menu can be useful to show what groups the server sees in the client user’s token.
- The “Clear Cached…” menu is added to the console root node. It is *STRONGLY* advised that this menu not be used.
- The “Agent Metrics” property page is added to the set of pages available for agent properties (or managed host properties for locally managed hosts.) This page provides a view of some useful agent performance metrics. Agents update this view every 4 minutes, when they renew their leases. In version 2.1, this view is replaced by the more useful Agents View, available as a right-click from the Managed Hosts node.
- The “Indexer States” property page is added to the set of pages available for agent properties (or managed host properties for locally managed hosts.) This page displays what indexers are in what states, and is used to calculate the “Data State” column on the managed hosts view.
- The “Clear Resource Activity” menu is added to the list of menu items for each managed host. It is *STRONGLY* advised that this menu not be used.
- When adding a cluster managed host, the checks for the validity of the cluster can be skipped. This allows the addition of cluster hosts in some cases where the native validation APIs report erroneous data.
    - NOTE: This tweak can be used to add hosts to QAM which are not clusters. It is strongly advised that this approach not be used. This method of host addition is only intended for virtual cluster nodes not having a computer account.
- Exceptions and other errors displayed by the QAM console are rendered with additional detail, including stack traces and other properties. This can be very useful to development when diagnosing problems.

Scanning Threads

In some cases, it can be desirable to have the QAM agent use more or less scanning threads for its file system indexing operations. By default, the agent will use 4 threads. However, on some very powerful servers, with high throughput IO subsystems, performance increases in indexing performance can be achieved by increasing this value. Conversely, on older servers whose IO subsystems are not capable of servicing significant amounts of throughput, lowering this value can decrease the impact QAMs agents have. The value can be set by creating a REG_DWORD registry value called “**scanningThreads**” beneath the following registry key:

HKEY_LOCAL_MACHINE\Software\Quest Software\Broadway**Agent\Services\fileSystem**

Note that all values in bold above are case-sensitive.

Agent working directory

In many cases, it can be desirable to have the agent maintain it’s various files on a disk other than that it was installed on. This is possible by creating a REG_SZ registry value called “**rootDirectory**” beneath the following registry key:

HKEY_LOCAL_MACHINE\Software\Quest Software\Broadway**Agent\System**

Note that all values in bold above are case sensitive.

When this registry value is present, upon restart, the agent will create it’s files in a sub-folder of the specified path, named after the agents instance name (BroadwayAgentService for local agents, BW_<guid> for remotes).

Warning: Upon uninstallation, any subfolders or files within the folder specified by the rootDirectory value will be deleted. Ensure that no information from other applications are stored in this directory!

Diagnostics and Troubleshooting

The following are a couple of common situations which occur during QAM deployment:

Agents sit in a “Waiting for First Connect” state forever

This occurs when the agent is unable to connect to the QAM server. The following are a few causes of this issue:

- The Service Connection Point objects for the domain in which the agent resides are not present. This can be fixed by examining the properties of the Managed Domain in which the agent exists, and forcibly creating the SCPs.
- The URL to which the server connects, as defined in the serviceBindingInformation properties defaultUri value, cannot be contacted from the agent host. This can be tested by copying out the URL (http://<server>:8721/broadway/indexserveragentport), and trying to contact it through Internet Explorer on the agent machine. If a web page is displayed, the agent should be able to properly connect to the QAM server.

Agent installation fails

There are a number of reasons an agent could fail to install. One of the easiest ways to see what has occurred is through the “Configuration Message” property of the agent on its details properties page. This usually indicates what kind of issue caused the installation to fail.

Agent data state sits in the “Performing Initial Scan” state for a long period

During full file system security scans, agents will switch their data state to “Performing Initial Scan”. They remain in this state until all security indexers (the file system, windows service identity, and share\group) have completed. In some cases, if one of these scanners takes a very long time, the data state value can persist in the Performing Initial Scan state for some time. The easiest way to determine how long this state will persist is to look at the Agent Metrics view for each of the agents targeting the managed host in question. The “Items Enumerated\sec” value in this view shows the average performance of the file system indexer through it’s most recent full scan. If this value is less than 600 items per second, the cause of it’s slowness should be investigated. This information is integrated into the Agents view present in the 2.1 release. Generally, problems with this value can be attributed either to latency between the agent host and it’s target, and disk IO slowness on either the targeted managed host, or the agent host itself.