MSP BUILDER

Tools for MSP Success

RMM Suite for Kaseya VSA
Operations & Customization Guide

Core Automation

MSP Builder, LLC
Version 2.5 / Release 21-060
Glenn Barnas

Last Updated: 2021/06/21

MSP Builder RMM Suite for Kaseya VSA

The MSP Builder RMM Suite contains proprietary software, including unpublished source code. All software is (and remains) the property of MSP Builder LLC and no transfer of ownership is granted or implied.

The MSP Builder RMM Suite software is designed to audit, monitor and manage computers that use a Kaseya Agent application. It is not designed or configured to collect personally identifiable information and should not be configured to do so without the consent of the individual or to be used in any unlawful manner, or in a manner that requires the consent of an individual.

MSP Builder LLC
385 Falmouth Ave
Elmwood Park, NJ 07407
201-300-8277

Contents

Introduction

The MSP Builder RMM Suite for Kaseya VSA provides Kaseya users with an advanced collection of components that highly automate the operation of a VSA platform. With these components in place, MSPs no longer need to manually evaluate systems to determine configurations and apply monitor sets. The use of agent templates, which are generally inflexible, is eliminated. From the time an agent first checks in, automation is applied to perform agent customization, deploy support tools, and detect monitorable items & apply the appropriate monitors.

Most agents require little if any customization – a typical customer onboarding consists of discovery, agent deployment, and confirmation that all devices were discovered. Only the servers require a manual process to determine any interactions and establish a patch/reboot sequence. Even with all of the automation, systems are easily customized at the customer, group, or agent level.

The RMM Suite allows you to do more with higher accuracy, without increasing resources.

Overview

The RMM Suite's Core Automation consists of the following standard components.

Audit Tool

The MSP Builder Audit Tool is a key component of the RMM Suite that identifies the applications, roles, and features located on every agent. This information is returned to the VSA to control the deployment of monitor sets specific to the roles found on the system. This tool runs quickly at least once per day to detect changes to an agent's configuration. These changes are reflected by monitor sets being added or removed without any manual intervention.

Agent Monitors

A collection of highly tuned monitors is a core part of the EMM Suite. The monitors employ standardized content and format to allow automated processing and parsing, not only in the MSP Builder Intelligent Ticket Processing System, but in any external ticketing system such as ConnectWise or BMS. The monitors are used to check system services and Event Logs and are highly optimized to focus on events that can be acted upon. This eliminates hundreds of help-desk tickets that result from the commonly used sample monitors. This "noise" is removed from the monitoring system, allowing MSP Engineers to focus on resolving the genuine issues.

Views

An extensive suite of VSA Views are included with the EMM Suite. Many of these are available to the VSA engineers to filter results and search for common conditions. They also form a standard format for creating additional views to meet an MSP's specific needs.

Additional views are also provided to control the deployment of System Policies. These "X" views are not globally visible to all RMM users but restricted to the RMM System Administrator or Master accounts. "XAPC" views control the application of "Auto-Pilot" policies that deploy monitors, maintenance, patching, and application updating with little or no manual configuration. The "XARC" policies control the application Auto-Remediation policies. These views can be copied to create custom automation solutions where required.

System Policies

Core Automation uses System Policies extensively to ensure that desired configurations are applied to systems that register with the VSA. These policies apply monitor sets, deploy and schedule patching and application updates, and can implement daily maintenance and Smart Monitors from our EMM Suite. This minimizes the effort needed to deploy these configurations, requiring only that the agent be placed into a specific machine group that has the policies linked to it.

Policies can be applied to groups, limited by filters (views), or applied to a specific agent for the utmost in configuration flexibility. Policies are even used to automatically remediate certain conditions, such as insuring that applications are maintained at a specific minimum version.

System Policies are a key part of Core Automation, which ensure consistency in the configuration of monitors and agent settings while reducing the manual effort traditionally required. A large set of "Auto Pilot" policies are included, which apply standard monitors to all systems automatically. A set of Extended Monitors are part of the Auto-Pilot suite which add very focused monitor sets for specific system roles, such as DNS, DHCP, SQL, Exchange, and several other applications. The daily audit procedure is fully customizable and comes pre-configured with over 60 unique system roles defined.

Patching Policies

Core Automation includes a complete set of patch policies that simplify the deployment of system updates. It does this by first breaking them into deployment groups that are applied to all systems, then workstation or server platforms, and then more specific groups that block patches based on specific conditions. These policies create a solid starting point for customizing a robust and reliable patching process. System Policies complete the package with standard patch deployment configurations to cover a wide range of conditions. These can also form the basis for even more custom configurations.

At this time, Software Management does not support Export/Import of its configuration policies. A recommended method of policy configuration is provided in this guide for manual setup.

Procedure Library

A diverse set of over 300 procedures for deploying and customizing applications, performing diagnostics, tweaking system settings, and similar support tasks is included with Core Automation. A special set of procedures utilize Ninite Pro (licensed separately and directly by the MSP) or Chocolaty to install, remove, or update a broad set of applications. Many of the procedures can be used interactively or as scheduled events for ongoing updates and administration. Additional procedures are available in the EMM Suite that target daily maintenance and use of Smart Monitor tools.

Network Monitor

Network Monitor significantly augments the Kaseya VSA with Out of Band (OOB) monitoring. Core Automation leverages two key monitor types in Network Monitor – Performance and Operational Availability – by providing a collection of monitor templates for both.

Core Automation performance monitors are tuned to meet several tiers of system capabilities. This allows the monitors to be applied appropriately and minimize the noise that results from the sample performance monitors, which are often based on manufacturer's reference architectures.

With the use of MSP Builder's Intelligent Ticket Processing, the performance monitors can be further configured to only alert within specific hours of operation (ignoring high-load times when backups or updates are done) or to never alert but only track performance. This allows the system performance to be assessed and system optimization tasks performed before allowing performance issues to generate alarms.

Operational Availability monitors perform specific tests that interrogate a server in such a way that it verifies that a service or application is functioning. Typical tests simply verify that the service or application is running, which cannot detect when a system is running but not responding properly. Core Automation includes monitors for Active Directory, DNS, SMTP, HTTP, and many other common system services.

The primary Operational Availability alert communicates with the server to assess general availability. This functions even when the server is under high load (response may be delayed slightly, but still responds to the query). This provides a high degree of accuracy for reporting a server-down condition. The alert triggers after 15 minutes, to allow time for a reboot to complete without unwanted alerts, although this time is easily customized. The more common method of reporting when the agent doesn't check in is highly susceptible to false alerts, as the agent is designed to "go quiet" during times of high CPU load.

Core Automation Utilities

The following is a table of Utilities (BMS Scripts), mapping the file name to its purpose.

RMAELR.BMS	Audit	Event Log Reporter
RMAELS.BMS	Audit	Event Log Scanner
RMASDU.BMS	Audit	Daily Audit Utility
RMASTR.BMS	Audit	Scheduled Task Auditing
RMASVC.BMS	Audit	List Services and Accounts
RMESMSG.BMS	Utility	Send GUI Message Utility
RMESTM.BMS	Utility	System Tray Utility - Maintenance Interface
RMUAAC.BMS	Utility	API Diagnostic Utility
RMUAAR.BMS	Utility	Local Admin Account Reporting
RMUAIB.BMS	Utility	Agent Init & Branding Utility
RMUATST.BMS	Utility	Alert Test Utility
RMUDFS.BMS	Utility	Disk Free Space Check (for pre-install checks)
RMUDVR.BMS	Utility	Disk Volume Report Utility
RMUGET.BMS	Utility	Get RMM Files Utility
RMUGOR.BMS	Utility	Get ODBC Info Utility
RMUINI.BMS	Utility	Manage INI File Utility
RMUKAR.BMS	Utility	Agent Redeploy Utility
RMUNNU.BMS	Utility	Ninite Update Utility
RMUOBA.BMS	Utility	Agent Onboard Automation
RMUODJ.BMS	Utility	Offline Domain Join Utility
RMUSCD.BMS	Utility	Sync Computer Desc Utility
RMUSCI.BMS	Utility	Set CustID Utility
RMUSNAP.BMS	Utility	System Snapshot Utility
RMUSPM.BMS	Utility	Set Policy Control Utility
RMUSUL.BMS	Utility	User Logoff Utility
RMUUAM.BMS	Utility	Local User Management Utility
RMUWAV.BMS	Utility	License Auth Init/Verify Utility
RMUWIFI.BMS	Utility	WiFi Get/Set Utility
RMUW32.BMS	Utility	W32Time Config Utility
RMXSVC.BMS	Utility	Service Remediation Utility

This Page Intentionally Left Blank.

Operation

This chapter will provide information on the day to day operation of the RMM Suite and Core Automation components. Some information will be in summary format, with details in a separate chapter.

Overview

One of the most important benefits of the RMM Suite is its extensive use of standards for operation and organization. This ranges from the naming of machine groups to aid in deploying policies to standardized formatting of event notifications, to the directory structure defined within the Kaseya working folder. While the specific standards documented here are our defaults, the key is to define standards that work for your organization, implement them, and *follow them!*

The rest of this chapter will cover the typical operation of the Core Automation components from a day to day use perspective, as well as providing insight into best practices for their use, starting with basic configuration settings.

Organization of Agents

One of the key elements of the monitoring platform should be a well-organized structure that clearly identifies the customer, host class, and location of the monitored device. Many MSPs lump all of the agents in a single group and use Views to filter the machine type. While this works quite well for manual operation and organization by the MSP engineers, the views are not available during event intake processing. The identification of customer, machine class, and location must be extracted from the machine group data. This will also allow systems running on desktop platforms to be treated as servers. Creating separate subgroups for Workstations and Servers also ensures that manually applied policies cannot be applied to the wrong system type if the view filter is incorrectly defined.

The standard Core Automation configuration employs the structure shown here:

```
customer.r.class.location.machine
```

- **customer** – The identity of the client receiving the monitoring services. We recommend that this be 15 characters or less in length and match the Customer ID in the PSA system that tickets are submitted to. A special customer group called “audit” is available that will prevent any automatic operations from being performed, with the exception of deploying the utilities needed for a detailed audit. This “customer” has multiple root-level sub-groups so that multiple customer audits can be performed at once.
- **r** – The root of the structure containing all agents associated with the customer. No agents are stored in this group, but it forms the base of all sub-groups and serves a critical role. This level allows a single customer to consist of both managed and unmanaged agents, organized by type and location.

  There are different forms of the root group:
  - **m** – Represents a standard, managed customer. A number or other qualifier can be added to define various levels of management. This requires customer-defined policies and is not a core feature. Other terms besides “m” can be used to define sub-organizations, especially if billing is performed separately.
  - **unm** – This indicates that the customer is “unmanaged”, and this prevents all of the Auto-Pilot policies from being applied. This means that monitors, patching, and application updates will not be performed automatically. If any of these features are desired, the appropriate policies will need to be linked manually. It is generally preferred to use a managed organization type and add restrictions than to manually add policies.

- **Class** – Defines the class of computer so that appropriate monitors can be applied. Our platform uses the following standard classes:
  - **servers –** Any machine operating in a server role, even if using a workstation platform.
  - **wkstns –** Any machine operating in a workstation role, even if using a server platform. For organizations with a large number of mixed platforms, sub-groups for PC, MAC, and LNX can be defined, although this is where we recommend the use of views.
  - **tclients –** Thin-Client systems. These usually require special procedures to unlock the device before updating and then re-lock after completion. Placing this into a separate group makes this special handling easier to perform.
  - **special –** Any system requiring special handling or processing. These are often exempt from common monitors, alerting, maintenance, and/or updating. Moving an agent into the Special group will automatically prevent all automation *except* for Daily Audit and Patch Scans.
- **Location –** A code that identifies the site. This is based on the United Nations LOCODE standard which provides unique codes for each city worldwide and employs an alternate method to identify cloud and hosted datacenter providers. The MSP is free to choose an alternate naming method, but using some form of location groups is strongly recommended.

Adapting the UN LOCODE Standard for Site Naming

A common challenge faced by all IT managers is the identification of a site by using a code. Some methods use an abbreviation of the city or the IATA Airport Code. This might work well for organizations with just one or two locations per state or region, but a company with many small offices in a single state or region could pose a challenge. There's also the issue of similarly abbreviated city names – consider Paramus and Parsippany, both located a few miles apart in northern New Jersey – the 3-letter abbreviation for both could be "PAR"!

Fortunately, an international standard exists that requires no stress or extra effort on the part of the IT management staff. The United Nations has published a directory of unique identifiers for almost every city in the world. The Paramus/Parsippany challenge is eliminated – Paramus becomes "US PRM" and Parsippany becomes "US PPY".

Machine Group Name	Parent Machine Group	Default
mspb.m		Yes
mspb.m.servers	mspb.m	No
mspb.m.servers.nj-ldo	mspb.m.servers	No
mspb.m.special	mspb.m	No
mspb.m.wkstns	mspb.m	No
mspb.m.wkstns.nj-ldo	mspb.m.wkstns	No

The UN LOCODE Directory for the USA and several other countries is supplied in XLSX spreadsheet format as a convenience to our customers, and other regions can be added upon request. This spreadsheet contains the original UN data, plus two columns labeled "RMM 6" and "RMM 8". These columns contain 6 and 8-character formats that can be copied and pasted directly into the VSA's create machine group dialog box. The two formats are slightly different:

- RMM 6 — This format uses the raw LOCODE value of Country and City, delimited with a tilde (~). The USA, having nearly 20,000 entries, has a slightly modified version of the RMM 6 code, which uses the State abbreviation and the City code, delimited with a dash (-). This improves recognition of State/City identities.
- RMM 8 — This format uses the two character Country code, a two character region/state code, a dash, and the three character city code. This provides complete country, region, and city identity for when an MSP or Corporate user supports locations around the world. *This is our recommended format and is used by our tools.*

For organizations with several locations in the same city, a short suffix can be added to the location code - we refer to this as a "qualifier". For rural areas that do not have a LOCODE identity defined, we recommend that you use the closest location that does have a LOCODE defined or define a custom code after verifying that your custom code is not already in use.

Third-Party Datacenter Locations

When customer physical equipment is hosted at a third-party facility (not a "cloud" provider" with virtual servers), the same naming format is used with the addition of a specific suffix. The location is appended with an underscore and a two-character provider identifier. This helps differentiate between customer owned and third-party hosting locations.

Identifying Cloud Datacenter Locations

With the advent of cloud-based computing, the concept of a city-based location may not be valid. A different format is used to identify third-party data centers where equipment is hosted. The datacenter is often associated with the customer's primary site, but has a "qualifier" code added.

The location code employs a suffix that identifies the hosting provider. The suffix starts with underscore, followed by the two-character identifier. Examples of data center IDs include:

- AW – Amazon WebHost
- AZ – Azure
- GG – Google
- RS – RackSpace

Thus, a site representing systems hosted by an Amazon data center for a customer located in NYC would be named "usny-nyc_aw".

These codes are not tied into the RMM Suite but offer a standard method of site identification. Feel free to adjust this model to suit your specific needs and accommodate local and regional service providers.

The RMM Suite provides automation that will look up the correct LOCODE information and create the client org and machine-group structure in VSA. A simple Excel Spreadsheet is used to define the client ID, name, and location info and the tools do the rest, resulting in a consistent structure for your client organizations.

The Kaseya Working Directory Structure

The Kaseya Working Directory is a folder on the agent system that Kaseya uses for staging and operation. When Kaseya is installed, the default folder is called "kworking". The RMM Suite defaults to use this standard location.

> **NOTE:** The Kworking structure exists for a specific purpose and not as a "catch-all" folder to hold downloads and other temporary files. All MSP Builder procedures use the system Temp folder location for this purpose.

None of the RMM Suite's procedures have this location hard-coded, so the standard "kworking" folder or any alternative may be used. If a custom folder is desired, the Global Setting Policy "Agent: Settings - Windows" must be edited to reference the desired Kaseya folder name. *We do not recommend changing this folder if your VSA is hosted (SAAS). These platforms do not have the ability to set the default working folder during agent installation and this can result in having multiple directory folders active.*

The procedures that are part of the RMM Suite will use the system temp location instead of the Kaseya Working folder for deploying applications or collecting data. This allows the temporary files and folders to be easily cleaned up after they are no longer needed, and prevents the Kaseya Working folder from being filled with unnecessary and often outdated files and folders. This also serves to improve system security by removing potentially sensitive commands and log files after they are no longer needed.

There are three locations (including two new folders) that Core Automation procedures utilize:

- The root folder itself – used for holding configuration files used by RMM Maintenance and other add-on utilities such as the log management service.
- BIN – a subfolder that contains the scripts, programs, and support utilities used by the RMM Suite. This folder is added to the System PATH variable, making these tools available from the command line. MSPs are encouraged to add their favorite support tools to the deployment folder to make them available on all client systems.
- LOGS – a folder where the RMM Suite tools write their log files. By using a standard subfolder for all logging, it is easy to Zip and collect all log files in a single operation. A procedure is provided for just this purpose.

Kaseya will create several additional subfolders for its use. Their presence will depend on the components deployed. They are not used by the RMM Suite.

The "RMM Admin" Account

MSPs generally create an administrator account that will become a member of the local administrator group of every agent. This actual account name varies by MSP, but is *referenced generically as the RMM_Admin* account in the RMM Suite. Several procedures exist to create, change the password, and remove this account, and these credentials are defined by Managed Variables. This allows the credentials to be set globally or per-customer as needed.

For optimal security, this account should be a member of the local Administrators group, should NOT be a domain admin account (use a separate account for this role), and should be different (at least) between the MSP organization and its customers. Ideally, a unique password will be set for each MSP customer.

An optional tool is available for download from our website that will encipher the passwords stored in the managed variables. The utility that sets the credentials will automatically detect when the credentials are ciphered and decipher them using the appropriate, customer-specific cipher key-part. This tool has its own Operation Guide.

The use of the local *RMM_Admin* account is not mandatory and is not used by any RMM Suite tool. The creation of this account is controlled by the Managed Variable RAUserID/RAPassword. If these are not defined, the local *RMM_Admin* account is not created.

Managed Variables

Many of the RMM Suite procedures utilize managed variables so that MSP-specific data can be defined in a single place. The variables can be accessed from the Agent Procedure / Schedule/Create menu. The standard RMM Suite variables are defined here:

MGUIArgs

This variable is optional and contains arguments passed to the Maintenance GUI. Most commonly, this causes the interface to be launched silently rather than displaying briefly.

NiniteCache

This is a customer / site-specific value that contains the UNC path to a local shared folder. This network share is used by NinitePro to cache the install files. The first time a new version is found, it is placed on this share. All other NinitePro updates will then obtain the software from this local share.

RAUserID & RAPassword

These are the name and password of the local RMM Admin account. The Init and Offboarding procedures create and modify or remove this local Windows account. It provides a standard account ID with local administrator rights. There are also procedures to update this password from the managed variable.

CAUserID & CAPassword

Similar to the RMM_Admin credentials above, these are used to set a Customer Admin account and password. These values should be unique to each customer! This account is used when a customer requests a local admin account for their own internal administration tasks. The Init process will create this account if the variable is defined.

NMUserID & NMPassword

These credentials are deployed to servers that are monitored by Network Monitor. A specific procedure is used to deploy this account where needed.

Custom Managed Variables

This would be a good time to document any custom Managed Variables unique to the MSP practice.

Special System Settings

There are several system settings where Core Automation utilizes specific configuration settings within the VSA. These are summarized here.

Organization Setup

- The organization ID for customers should be less than 15 characters whenever possible and should match the customer ID defined in any external ticketing system.
- The Organization Name used in VSA must match the name in the PSA in order to properly associate tickets generated by alerts with organizations. The VSA Org Name should be updated to match the name in the PSA if they do not match.
- Three (or more) setup types should be created to define customer alerting windows. These define the times when *monitoring* is permitted to send a paging notification for high-priority events. *It does not reflect when a customer can call for after-hours assistance or when monitors operate*. The Description field defines the hours of monitoring for weekdays and weekends/holidays using the format "WD-hh:mm/hh:mm,WEH-hh:mm/hh:mm, where "hh:mm/hh:mm" represents the start and end times of coverage. For 9 AM to 5 PM, specify "09:00/16:59".
    - MB-Standard_Coverage for typical weekday coverage.
    - MB-Extended_Coverage for Monday-Sunday, 6am to 10pm type coverage. Various levels of extended coverage may be used for Mon-Fri, or different hours of coverage, depending on the MSP service offering. Ten levels are supported by the Ticket Automation Suite, the default "Customer-Extended_Coverage" and "Customer-Extended_Coverage#", where "#" is in the range of 1-9, A-Z. These provide alternate schedules, such as Monday-Friday or extended coverage in alternate time zones.
    - MB-Full_Coverage for 24x7 coverage offerings.
- Every organization should be assigned a "Set-up Type" based on their coverage as defined above. This "setup type" defines "in coverage" times for after-hours notifications.

User Setup

The standard RMM Suite implementation suggests the following user roles:

- NOC-0-Support – An access level used by non-technical staff to maintain certain customer settings and review licensing counts.
- NOC-1-Technician – The default access level for all technical users. Provides access to all agent administration and support tasks
- NOC-2-Technician – Provides additional access to configure components such as Antivirus exclusions, backups, or other customer-wide settings.
- NOC-5-Specialist-<role> – a level that is granted to specific MSP engineers and grants access to tasks that could modify select configuration settings for a single VSA component, such as Backup, Security/Auth Anvil, and Patching.
- NOC-9-Mgmt – This level grants access to all VSA internal configuration settings except user security.
- NOC-9-Mgmt-Security – extends all controls to the user, including management of user security. This is equivalent to the Master account access.

If the MSP grants portal access to customer technical staff, then three additional roles are used.

- CUST-RD – Grants access to remote desktop, live connect, audit information, and agent status information only.
- CUST-L1 – Grants access to *view* KAV, KAM, and Patch information in addition to above.
- CUST-L2 – Grants access to *manage* devices and features within KAV and KAM, deployment of patches, and execution of Agent Procedures from the _KShared folder.

These roles are not *required* by the RMM Suite, but strongly recommended standards that provide appropriate access to the VSA platform, ensuring security and operational reliability.

All users and customers should be made members of the Personal (or other, "MSP Technicians") scope. Users should NOT be granted access to the Master Role or Scope and thus the configuration of the Personal scope is required. The Personal scope cannot be renamed or deleted and is Kaseya's default scope for technician access.

Configuring Roles

The following method will help in appropriately setting the access rights for the new roles.

- Configure the NOC-1-Technician role first and apply basic security.
- Select the NOC-2-Technician role, copy the settings from NOC-1-Technician, then grant the additional access rights.
- Select the NOC-0-Support role, copy the settings from NOC-1-Technician, then remove the remote agent management rights.
- Select the NOC-9-Mgmt-Security role and grant ALL rights. You may optionally remove rights for components that you don't wish to display, such as the intro or summary pages or components not used by the organization. We also strongly recommend removing all ability to connect to or manage agents, run procedures, or perform other agent management.
- Select the NOC-9-Mgmt role, copy the settings from NOC-9-Mgmt-Security. Edit the NOC-9-Mgmt role and remove the User Security rights.
- The various specialist roles, if used, can be copied from the NOC-1-Technician role and then adjusted to grant the required rights to administer the assigned VSA role(s).

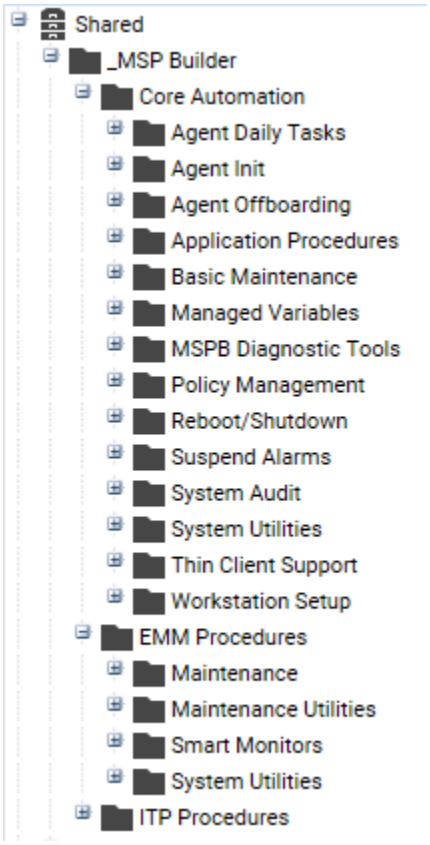
Agent Procedures

The RMM Suite includes a wide variety of ready-to-use procedures. All can be used without modification, and most will provide the basis for creating other customized versions to meet nearly any requirement. All procedures are located in the _MSP Builder root folder. There are three sub-folders therein.

- Core Automation — The core suite of procedures, usually shared to all employees/customers.
- EMM Procedures — Enhanced Maintenance and Monitoring procedures.
- ITP Procedures — Remediation procedures used only by Intelligent Ticket Processing.

Overview

Following the standard practices guideline, the Agent Procedures all share several best-practice methods.



- Each procedure has a platform prefix identifying where it can be used – ALL, WIN, LNX, or MAC.
- Every procedure has the Summary Description defined, allowing a rapid identification of what the procedure does.
- Procedures include comments to describe each logical block of commands. This explains the purpose of the procedure commands for future enhancements and modifications.
- Based on procedure templates, they share many common processes, including use of managed variables and system temp locations. By employing the same basic code for common steps, if a problem is found and resolved, the same updates can be applied to similar procedures without additional diagnostic effort.
- Procedures should be able to be created by any VSA user, but should not be enabled until reviewed and approved by a senior VSA administrator to ensure that the procedures meet the MSP's standards for documentation, security, and format. Procedures may be executed from a user's personal folder after approval for testing, but should be moved to a public location after verification. Maintaining private procedures is strongly discouraged.

Updating

The RMM Suite uses an automated update process to maintain the procedures, views, policies, and other components. Since these updates occur automatically, it is essential that you do not modify or customize any MSP Builder component! You can copy these and make changes to your copies, but if you modify our basic components, the changes will be lost during an automatic update.

We create an "MSP Customized" folder whenever possible to hold your custom components. When folders are not available, we suggest using a unique prefix, such as "MSP_" for monitors and "YAPC" for policy views.

MSP Builder will not be responsible for the loss of customizations made directly to our components as the result of an automated update process!

Core Automation

The procedures located in the Core Automation folder represent general-purpose tasks that can be applied to any customer or host system. (Some procedures are specific to servers, workstations, or thin-clients and are so marked.) These procedures should not be modified except where instructed to assure proper operation of all components.

Agent Daily Tasks

Platform-specific procedures that deploy and execute the MSP Builder tools. These deploy and update agent tools, initiate daily Maintenance and Smart Monitor applications, and run the Daily Audit.

Agent Init

Performs all the new-agent check-in configuration and customization tasks. Thin Client platforms are identified and the FBWF is disabled prior to initialization.

If the MSP wishes to perform additional customization tasks when a new agent checks in, they should use the RMUOBA.BMS application run from the **ALL-Agent Onboarding - 3 - MSP Customization Tasks** procedure. This is invoked after all MSP Builder initialization tasks have completed. See the Administration section for details on configuring the Onboarding Automation app.

Agent Offboarding

This contains a procedure that should be run when a machine is to be decommissioned. It removes all logs, registry settings, procedures, scripts, and other potentially licensed applications, and then removes the RMM_Admin user account.

Application Procedures

Most of the procedures in this folder are related to the installation and/or customization of applications. Some procedure groups of interest include:

- App Installers – a collection of procedures for installing applications on various platforms. These depend on external installer files deployed from the VSASharedFiles\Applications folder.
- App Management – procedures that interface with NinitePro or Chocolaty to perform installs, removals, and updates of supported applications. This requires a separately licensed version of NinitePro (command-line classic) if Ninite is to be used.
- Office 365 – a collection of configuration files that download and deploy various versions and configurations of Office 365 products.

Basic Maintenance

Procedures for simple system maintenance, primarily for Mac and Linux platforms.

Managed Variables

Procedures used to load Managed Variable data without causing errors when data is not defined.

MSPB Diagnostic Tools

Tools used by MSP Builder support to perform diagnostic and remediation steps.

- **Verify WMI for KNM** will configure WMI security on the host where it is run to allow Network Monitor to perform WMI queries. This procedure is run when the monitor reports a failure to communicate with WMI to correct the security settings.
- **Verify Auth** helps identify problems with RMM Suite license authorization.
- **Agent ID Validate** finds agents with invalid MSP keys.
- **Agent Redeploy** removes and re-installs an agent using the proper MSP key, or moves an agent from one VSA platform to another.
- **API Test** verifies that our tools can authenticate to the VSA API interface.

Policy Management

These utilities add or remove "Policy Blocker" values from the Policy Control custom field. These values control whether certain MSP Builder operations and monitoring will be performed and give the MSP the ability to easily override automatically configured tasks on a per-agent basis.

Reboot/Shutdown

A small collection of procedures to perform system reboots on any supported platform. The reboot procedures all suspend alarms for 30 minutes to prevent unwanted alerts. There are two Reboot - Patching procedures that will check the patch code for a "-R" suffix. If this is found, the procedure will not reboot the agent. This allows these procedures to be called from all server patch schedules.

Two procedures are also provided to allow or block access to the shutdown link on the start menu of Windows PCs. This might be used when users click shutdown each night instead of logging off. Removing the shutdown link makes the logoff link the default action. This may be needed to ensure that maintenance actions are performed each night.

Suspend Alarms

These procedures simply suspend alarms for a specified period, as well as one procedure to un-suspend any suspended alarms. These are usually called from other custom procedures but can also be invoked directly when needed.

System Audit

These audit procedures gather specific information from the target system and update custom fields or upload configuration files extracted from agent settings. These procedures run during the initial agent check-in and once a month thereafter. Servers run the audit during the first 3 days of the month while workstations run the audit during the middle of the month. (These are separate from the standard Kaseya audit process, which runs independently on a weekly schedule.)

The Data Query procedures are stored here and are invoked daily after the maintenance tasks. These collect data including that which is generated by the maintenance procedures and update the custom fields for each agent. The values obtained can be customized to update additional custom data fields. This information is often volatile or represents information used by Auto-Pilot policies that must respond to changes in system configuration. Adding a new role to a server, for example, will be detected within 24 hours and the appropriate monitoring and other Auto-Pilot controlled procedure will take affect within minutes of detection. The Daily Maintenance tool is highly customizable - see the configuration section for more details on collecting custom data values.

System Utilities

A large assortment of procedures that are used for typical System Administration tasks. These procedures can report status, send a user a message and optionally return a response, notify an admin (via email) when an agent comes online, and control various maintenance actions.

Thin Client Support

A set of procedures that help in the setup and configuration of Thin Client workstations.

EMM Procedures

The procedures in this folder comprise the Enhanced Maintenance and Monitoring components. These are invoked by the Core Automation's Daily Tasks.

Maintenance

The Maintenance procedure initiates the platform-independent maintenance tool. Maintenance tasks run based on the configuration files that were deployed to the agent, and by the EMM configuration settings.

Maintenance Utilities

A collection of procedures to support, configure, and validate Daily Maintenance operations. One of the most important procedures here is the **Upload All Maint Logs as Zip** procedure, which is used to collect all configuration and log data and is essential in troubleshooting RMM Suite issues.

Smart Monitors

This folder contains the Smart Monitors, including the daily Smart Monitor suite and the monitor used to detect the transition of Internet service from primary to backup and then the restoration of the primary service link. The procedure deploys and configures the monitor, or it can be used to simply update the configuration. The Internet Gateway Failover monitor must be deployed to a specific machine where the tests will be performed for the client.

System Utilities

Utilities used by the EMM components are located here. The Deploy EMM Config Files procedure deploys the Maintenance and Smart Monitor configuration files to the agent.

ITP Procedures (Intelligent Ticket Processing)

Procedures in this folder are used by the Advanced Ticket Automation Suite.

Auto-Remediation

These procedures are used by the Auto-Remediate logic in ITP. These procedures should generally not be run manually as they expect to have specific data provided to them from the ITP service.

The ITP can invoke *any* agent procedure from any folder, and in fact usually runs several standard procedures. The only distinction is that the procedures in this folder expect data to be provided by ITP to know what tasks to perform and thus are not suitable for manual operation.

Discovery

The Discovery module is used to identify network devices on a specific network subnet. It is used by the Kaseya Network Monitor module, so proper configuration of the Discovery module is essential for proper operation and integration.

Overview

Core Automation utilizes Discovery to identify servers and network devices that will be monitored by Network Monitor. Each subnet must have a computer with a Kaseya agent installed to allow discovery operations. When customers have multiple locations, they are treated as unique subnets. Each location and each subnet at a location must be defined separately.

Setting Up Discovery Options

The key to a successful discovery operation is the proper core configuration. Certain Discovery features that are designed for managing a single in-house network will cause issues when used to monitor multiple client networks. Another key task is to define unique and meaningful network names that include the customer ID. This is essential for properly identifying network gateway failures.

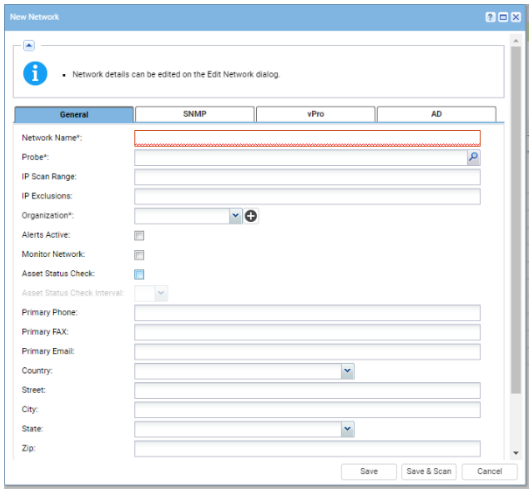
- In the Administration / Settings page, set:
  - "Enable automatic network harvest" must be **OFF**. Enabling this option in an MSP configuration supporting multiple clients will rapidly create an unusable discovery configuration. This option should only be used for single client (in-house) implementations where all networks belong to the *internal* customer.
    This option has been deprecated and is no longer available in VSA 9.2 and above.
  - Other discovery settings can be configured as needed, but default settings are acceptable. The RMM Suite does not utilize any of these specific settings.
  - Alert Defaults can be defined as needed but are not used/needed by the RMM Suite.

When the discovery module is configured properly, only one network per customer network (subnet) should be displayed, and only one "noNetwork" location should exist (Kaseya 9.1 and earlier). With release 9.2 of VSA, the "noNetwork" should not be present when properly configured. If multiple or duplicate networks are currently displayed, all of the devices should be deleted (Discovered Devices – Grid View; select all; Delete), then the network(s) should be deleted and then recreated using the process below. It's much better to start with a clean slate when working with Discovery.

Configuring Networks for Scanning

To configure a new customer network for a discovery scan:

- Navigate to Networks - By Network.
- Click the New button.
- Define the network name (see Network Naming below for format information).
- Select a probe system. A probe must have a Kaseya agent installed.
- Define an IP Scan Range. This range must be within a network subnet reachable by the probe and is used only when scanning part of a network or multiple subnets. This field is generally left blank and auto-populated by data from the probe's network interface.
- Select the organization id associated with the network.
- Select the "Monitor Network option if KNM monitoring is desired. Alerts Active should be OFF.
- All other fields on this panel are optional.



Click the Save button to scan later, or the Save and Scan button to save the configuration and immediately initiate a scan.

If you wish to automatically deploy agents to supported systems, use the Save option and then configure the Agent Deployment Policy. Perform a scan once the deployment policy is defined.

- Select a network from the discovery By Network screen, then select the Agent Deployment Policy table and click Edit.
- DO NOT check the Automatically Install… option(s) for Windows, Mac, and Linux platforms.
- Select the package to use for the install. The Default package is the preferred one to use.
- Select the machine group where the devices will be placed. Choose the specific customer, host-class, and location to allow proper onboard automation to be leveraged.
- Select the Designated Deployer Agent. This will be a system with a Kaseya agent that will be used as an intermediate deployment platform. The agent software will be copied from Kaseya to the deployer and from there to the target system for installation.
- Define the admin credentials needed to perform an installation. This should be a Domain account with local admin rights on every computer, or a local account with admin rights and consistent credentials on every computer.
- Select the servers from the list of discovered assets and click the Deploy button. Once these are complete, change the target install folder in the auto-deploy settings, select the workstations, and click deploy. While this is technically a semi-manual process, it allows the onboarding automation to properly trigger the appropriate tasks for servers, workstations, and client locations.

There are many things that can interfere with discovery and automated agent install. Some preparation steps include:

- Verify that you can log into every computer with a single set of credentials, and that account is in the administrators group of every computer.
- Disable any Internet Security software as well as Antivirus and Anti-Malware applications.
- If multiple subnets are to be scanned, ensure that Gateway Antivirus/Security is disabled, or use a deployment agent in each subnet. All subnets should be accessible from the deployment agent.

Network Naming

The name given to the discovery should follow the format used for machine group locations. This will aid in assigning the network to the customer in Network Monitor and provide name consistency across all modules. The use of this format is *required* to properly parse gateway failure events!

The network name format should be `customer.location[.VLAN]`

The “location” portion should match the machine group location ID. The VLAN portion is not used unless the location has multiple VLANs that contain devices to be monitored. Each VLAN subnet needs to be identified and scanned separately and requires at least one device with a Kaseya Agent installed.

Post Scanning Tasks

Once discovery identifies the devices on a network, all of the devices should appear in the Discovered Devices list. Use the “Network:” drop-down to select the network that was just scanned.

- Optionally, deploy agents to the systems without agent software installed.
- Identify and select any device that will NOT be monitored with Network Monitor. These devices can be deleted from the Discovered Devices list. Workstations are usually not monitored by Network Monitor. Depending on MSP policy and customer requirements, printers and some network devices may also be removed. This will limit the objects displayed in Network Monitor to only those that will be actively monitored with KNM.

Other Considerations

Core Automation does not utilize domain discovery features. Its use is left to the discretion of the MSP. We do suggest cleaning up the customer AD environment and removing stale objects prior to deploying domain discovery to avoid generating many “phantom” objects including agents and VSA users. Ideally, AD will be configured so that OUs are used to distinguish between workstations and servers, as well as the site where they are located should the client have multiple offices. This would allow Domain Watch to route agents into the correct RMM Suite group structure. *If AD is not configured in this way, it will not be possible to utilize Domain Watch as it will continuously move agents into a single group. This will affect if/when automation is applied.*

Scheduled network discovery scans should generally be avoided except – possibly – during onboarding of a new customer. The schedule should be removed once all devices are found. Using scheduled scans will result in workstations and other unwanted devices being discovered and appearing in the Network Monitor structure.

Using network discovery speaks volumes for implementing a well-designed address assignment process. If all servers are in a specific address range or subnet, then it becomes easy to scan for the devices to be monitored while eliminating the devices that simply contribute noise and disorder to the platform. We provide several Standard and Practice documents on our website that outline effective and efficient methods to design and secure customer environments and develop standards that aid automation.

Monitor

The Monitor section permits the definition and assignment of monitors, which generate the alerts when specific events occur. In addition to defining the actual monitor objects, the message delivered to the ticket can also be customized. Core Automation defines monitor sets here, but only applies them through System Policy-based automation.

Overview

There are four common types of objects that can be monitored – Services, Performance Counters, Processes, and Event Log events. (Mac/Linux systems can monitor specific logs for events as they don't have an "Event Log".) SNMP traps and a specific set of external system monitors are also supported with MSP defined monitors.

In an RMM Suite implementation, only services and event logs are monitored by this module. Performance, SNMP, and other external services are monitored by the Network Manager module, which offers better Out of Band monitoring capabilities. The use of process monitors is left to the MSP as these are generally application and environment-specific.

RMM Monitor Sets

The RMM Monitor Sets consist of a collection of Service monitors and Event Log monitors. Note that monitors are associated with machines via System Policy and should not be manually assigned here.

Service Monitors

The RMM service monitors provide a "core" set for servers and workstations, checking on the standard services that should be active on all platforms. Several additional monitor sets define services related to specific applications found on servers.

All service monitors associated with servers are Priority 1 and will result in a notification if auto-remediation is not successful. These monitors are identified as "actionable", which causes ITP to perform a remediation process when the alert arrives. The remediation process attempts to restart the service by issuing a Stop command, then it terminates any service still running, finally restarting the service anew. A single remediation task handles all service-related alerts.

To define a new service monitor:

- Prepare a Monitor Set ID. This uniquely names the monitor for processing by ITP.
    - Format is *name*.SVC.*platform.priority*.Act
      Name should define the monitor set; *Platform* identifies "s" for server or "w" for workstation; and *Priority* should be "P1" through "P3" depending on the severity that the service failure will cause.
- Navigate to Monitor / Edit / Monitor Sets
- Expand the Shared / RMM / Service Monitors folder. (Optionally, expand the Clients folder, open or create a client specific folder.)
- Click the "New monitor set" button
- Enter the Monitor Set ID in the Monitor Set Name field, and optionally (recommended) enter a description. Click Save.
- Select the Services Check button.
- Click the "Enable Matching" checkbox.
- Click Add.
    - Select a Service from the drop-down list, then enter a description.
    - Set the Re-start attempts value to 1 to allow the agent to perform the first restart.
    - Set the Restart Interval to 1 minute.
    - Set the Ignore additional alarms value to 15 minutes.

- Click Save.

The new monitor should be defined in either a new or existing System Policy. Note that workstation monitors are usually set to Priority 3 to prevent after-hours alerting.

Auto-Pilot Monitor Sets

The "Auto-Pilot" set of System Policies can apply specific monitor sets based on the roles found on a system. For optimal use, the monitor sets should be created for a specific role. Using the "SQL" role as an example, add all of the monitors related to MS SQL Server – including any server or instance-specific service names – to a single SQL monitor set. Turning on the "Enable Matching" option will ensure that only the matching services that exist on the system will be monitored. Note that there is nothing particularly special about an Auto-Pilot monitor set other than it being specific to a single product or role.

Process Monitors

Process monitors allow alerts to be generated when a process starts or stops running. Such monitors are generally very specific and as such, none are provided by the RMM package.

Monitor Set IDs are similar to the Service Monitors, except that the second field uses "PRC" instead of "SVC". Use a monitor type of Alm (alarm) unless a remediation procedure has been created and integrated with ITP. If a remediation process was created, the "Act" type of Monitor Set ID can be used.

The process can generate an alert when it starts or stops. In most cases, a custom process name will need to be defined as the choice of default monitors is quite limited. Refer to the Kaseya documentation to create a new process definition.

Performance Monitors

Performance monitors can generate alerts when specific monitors exceed a defined threshold. Core Automation uses Network Monitor for performance monitoring, and no pre-defined performance monitors are provided. The use of agent-based performance monitors is left to the discretion of the MSP.

Configuring Alert Notifications - the Alert Name Format

Core Automation, and Intelligent Ticket Processing in particular, depend upon a well-defined and standardized notification format. RMM Suite notifications are defined during installation and generally do not need to be changed. The format of the alerts is defined in detail in Chapter 4.

Most of the alert notification formats are defined by navigating to Monitor / Agent Monitoring / Alerts. The alert type is selected from the Alert Function drop-down. When the alert type is selected, the Format Email button is clicked and the Subject text is defined in accordance with RMM Suite requirements.

All alert notifications consist of six fields delimited with vertical bars (|).

- Field 1 – Alert Name.
- Field 2 – Alert Type (Alarm, Warning).
- Field 3-5 – Optional alert data, used to further qualify/classify the event.
- Field 6 – Monitor Set ID – a value consisting of 5 fields, delimited with a period. This is used for automated parsing of events in ITP and the ticketing system.

Defining Event Log Monitors

The Windows Event Log provides an excellent method of reporting conditions to Kaseya. Improper configuration of the Event Log monitors, however, can quickly overload the VSA with alerts.

When developing the RMM Suite components, our engineers looked closely at the standard event log data that was collected from the prior year. We found that almost 85% of the alerts generated had tickets closed with "information only" or "warning for unsupported condition". This represented considerable time for an engineer to review the alert and determine it had no value. The set of Event Log monitors

provided by Core Automation have been carefully crafted to only contain events with specific resolution actions. The event monitor sets have been further divided into separate sets representing the core events at both Priority 2 and Priority 3, plus separate monitor sets (with appropriate priority levels) for various optional applications. Further, every monitor has been configured with specific source and event ID values, and text filters to further restrict events that share source and ID values.

By configuring specific source, event, and description filters, wildcard monitors are avoided, and generally negate the need to define an "ignore" type of monitor. The more specific that the event monitor is, the less chance you will have of receiving false alerts.

Configuring an Event Log Monitor

Event Sets follow the standard Monitor Set ID naming format, using "EVT" as the type. Event Log Monitors can be actionable as well as request and regular alarming type. If an actionable monitor set is created, *all* the events in the monitor must be resolved by the same remediation procedure. These monitors usually have just one event, or possibly a small collection of related events.

To create a new event monitor set:

- Select "<New Event Set>" from the drop-down list.
    - Enter a valid* Monitor Set ID in the Event Set Name field and click New.
- Enter the Event Source ID in the Source Filter field. This can be obtained by examining the event log entry that you want to monitor. Enter the full source name – avoid using wildcards.
- Enter the Event ID into the Event ID field.
- Optionally, enter text from the body of the event log alert to further qualify the monitor. Place wildcard characters (*) at both ends of the text. Use a single wildcard character if no qualifying text is needed in this field.
- Place wildcard characters in the Category and User Filter fields
- Click Add to add the event to the monitor set.
- Repeat the above steps to add other, related monitors.
- When complete, click Deploy, then Close.

Associate the new Monitor Set ID with a new or existing System Profile, then link the profile to a machine group or individual machine. Event Log monitors can also be assigned to Auto-Pilot policies to create a complete Event and Service type of monitor.

Monitor Sets should never be applied directly to agents as System Profiles may cause them to be removed. Direct management of monitors and other agent configuration leads to inconsistent configurations and hard to diagnose monitoring problems. Always utilize and rely on system policies to deploy monitor sets.

*Monitor Set Names should consist of a short identifier, a dash, the event log source id (A=application, S=system, C=security) in the first field. The use of the event log source identifier assures that the monitor set is later properly associated with the correct event log when configuring the policy. Use a prefix in the name to group your custom monitor sets together and separate them from the RMM Suite monitors. Ours use a prefix of "MB-".

Patch Management

The RMM Suite supports patch management by providing a set of Patch Policies that are pre-configured for use in a typical MSP environment. The policies operate on an increasing denial method, with the most common policies approving the largest set of patches, and then denying specific groups of patches with "blocking" policies.

Note that Microsoft does not classify Windows 10 Upgrades as patching, and as such, these are not delivered by Patch Management. MSP Builder provides a procedure to perform these upgrades, which is discussed at the end of this section.

A computer is usually assigned to two base policies - _Baseline and either _Prod-Servers or _Prod-Workstations. This allows most updates while blocking a select few. Then, based on customer requirements, other patch policies are added (via custom System Policies) to prevent automatic installation of certain types of updates. An agent system can be a member of multiple Block groups to tailor it to receive only the updates it needs.

Review the Standard Policies

Start your implementation by reviewing the standard policies included with the RMM Suite. Identify blocking policies that agents should be members of and determine if additional policies should be created to block other update types.

_Baseline	This policy approves all patches and denies only those that should never be applied to any system (server or workstation). This policy usually denies only optional software such as add-on media or games.
_Prod-Servers	A policy that will deny only server-based patches that should never be applied. An example of patches that would be denied in this policy are updates for desktop type applications that would result in an installation of the application. Certain Skype updates are a good example.
_Prod-Workstations	Patches that are globally denied on workstation class systems are defined by this policy.

The remaining policies are used to block specific types or classes of updates.

Block All but Critical Updates	This policy allows only critical updated to be installed, denying all other patches.
Block All Optional Updates	This policy blocks the optional updates, allowing both critical and important updates to be installed.

The two policies above are particularly useful when starting a patch cycle in an environment that is severely delinquent. They will allow only Critical updates, then only non-optional updates, and finally all remaining updates as the agents are added to and then removed from these policies.

Block DotNET Install	All updates that install or upgrade DotNET are blocked by this policy. This may be important for customers with applications based on DotNET.
Block IE-## Install	Several similar policies that block the IE upgrade installers.
Block ?? Service Pack Install	Policies that are used to block various service pack installations, including workstations, servers, Office products, and applications.

Creating New Policies

New policies will need to be created when customer requirements demand that certain updates or update types should be blocked.

- Start by identifying the updates to be blocked. Decide on a name that identifies the policy based on the update types. Follow the standard by beginning the name with "Block".
- Enter the name for the new policy and click the Create button.
- Set the default approval status as needed. This will simplify monthly administration by auto-approving or denying certain classes of updates. Those marked "Pending Approval" will need to be manually reviewed each month and either approved or denied. Note that auto-approval applies only to newly released updates. Updates discovered on systems via patch scans will need to be manually approved regardless of the auto-approval (or denial) setting.

System Policies should be used to assign Patch Policies. When new patch policies are created, one of the system policies that defines patching configuration should be updated or duplicated and modified. Auto-Pilot policies are provided that deploy the most common configurations without additional effort.

Monthly Management Tasks

As new patches are released by Microsoft each month, they will need to be reviewed and then either approved or denied in each of the patch policies. It is possible that a new class of update will be released, requiring the creation of a new patch policy.

Approval by Policy

The easiest method is to review and approve updates by patch policy. This leverages the policies and auto approval settings to minimize the manual steps needed. There is generally no need to deny superseded updates as the system will only download / deploy the latest versions of updates. Only deny the updates related to the policy (DotNET for Block Dot Net Updates) or known to cause issues and approve all remaining updates.

- Navigate to Patch Management / Patch Policy / Approval by Policy.
- Select the first policy from the drop-down list.
- Check the Pending Approval column to review any updates in pending status. If there are a large number of updates to review, you can check and manage them by clicking the value on each classification line. For a smaller number of updates, they can all be reviewed by clicking the value on the Totals line.
    - Check each of the updates and select any that should be denied based on the policy title. (Even though DotNET might be denied by a policy, it should be approved in policies that don't specifically exclude DotNET.) Click the Deny button when all the denied updates (if any) have been selected.
    - Wait for all updates to be displayed (scroll bar stops moving), click the Select All link and click Approve.
    - Click "Back" to return to the Approval by Patch Policy screen.
- Select the next policy from the drop-down list and repeat the above step until all patch policies have had their updates approved or denied.

Approval by Patch

Approval by Patch is an effective way to deny a patch in all policies. This does violate the process of approving all updates in the core policies and then denying them in specific Deny policies, so this method should be used with caution. This method is not generally recommended as it can make management difficult to track.

KB Overrides

Similar to Approval by Patch, the KB Overrides option will approve or deny a specific update by KB number. This method should be used only when an update is to be denied to all agents, regardless of patch policy membership rules.

Using System Policies for Patch Management

An extensive set of System Policies are provided for patch management. These can be used as is or modified to suit the MSPs specific needs. The Auto-Pilot policies automatically configure the most common patch settings. The entire Patch Management (Auto Pilot) folder is linked to the Org Root.

- **Patching: (Auto) Wkstn - h:mm THU - <type>** – Schedules all workstations to be members of the core patch policy groups, controls the alerting and reboot configuration, and schedules weekly scans (Monday) and updates (Thursday). There are 4 schedules available, controlled by the name applied to the "wkstns" machine group. **"<type>"** is one of "Hold" or "Resume". The "Hold" option will *not* continue patching if the workstation was offline during the scheduled patch window, while "Resume" will continue updating the computer when it is powered back on. The user will be notified that patching is resuming.
    - **wkstns** - The standard workstation patching schedule and method.
        - Updating will be performed on Thursday between 00:10 and 02:00;
        - Patching will be performed on Thursday between 02:30 and 04:30;
        - Updating & Patching will NOT resume if the workstation is powered off.
    - **wkstns1** - The alternate workstation patching schedule and method.
        - Updating will be performed on Thursday between 00:10 and 02:00;
        - Patching will be performed on Thursday between 02:30 and 04:30;
        - Updating & Patching will resume if the workstation is powered off.
    - **wkstns2** - The late workstation patching schedule and method.
        - Updating will be performed on Thursday between 04:30 and 06:30;
        - Patching will be performed on Thursday between 06:30 and 08:30;
        - Updating & Patching will NOT resume if the workstation is powered off.
    - **wkstns3** - The late workstation patching schedule and method.
        - Updating will be performed on Thursday between 04:30 and 06:30;
        - Patching will be performed on Thursday between 06:30 and 08:30;
        - Updating & Patching will resume if the workstation is powered off.
    - When a "Resume" type of policy is deployed, the Maintenance Interface will detect the resumption and display a notification that patching is being done. This message will not be displayed if the user interface is in Kiosk mode or otherwise disabled.
- **Patching: (Auto) Wkstn - Server Schedule** – Overrides the default workstation schedule when a Server Patch Code is applied to the workstation. This causes the workstation to follow the server process - reboot/patch/reboot at a specific time unless the "-R" suffix is applied to the patch code.
- **Patching: (Auto) Server - Config Only** – This policy applies the common configuration settings to all servers, including a weekly patch scan. The actual patch installation is controlled by a separate policy. This policy will reboot the server after patch installation by a procedure that will suppress the reboot if the schedule code contains a "-R" suffix.
- **Patching: (Auto) Server - <schedule>** – Assigns a Patch Schedule based on a code entered into the "Patch Schedule" custom field. This schedule configuration disables alerts from the start of the patch cycle window, reboots the server, and then deploys the patches. Schedules are available for Every Saturday, the Third, Fourth, or First Saturday or Sunday, and select times throughout the day on Wednesdays. The pre patching reboot action defined in this policy will be suppressed if the Patch Code has a "-R" suffix.

Manually Linked Patch Policies

There are several policies in the Special Configurations folder to accommodate non-standard situations. These should be used only for special situations and can be linked to a customer machine group or directly to an agent as requirements dictate.

- **Patching: Wkstn - 2:30 THU - Excl DotNet –** A policy that uses an additional Patch Policy to exclude DotNet updates. This can be used as a template for creating other policies to exclude specific update categories defined by patch policy.

Controlling Workstation Rebooting

Rebooting of workstations is controlled by an external process and configuration file called PATCH.INI. It is located in the KaseyaShared\MSPB_EMM\Maint_Cfg_Common folder and is deployed and customized using the same methods as other Maintenance tools. This file is internally commented to document each of the settings. It consists of one section called PATCHING.

Note that patching is NOT considered part of the RMM Suite Maintenance tools, but has been integrated with the maintenance configuration process to leverage an existing and well-documented method of deploying configuration files. Unlike other configuration files that are used to override the default settings, the PATCH.INI file *must be reviewed and configured to meet the needs of the MSP and their customers.* The default settings may not provide the desired results.

Nag Messages are part of the User Interface, described in the Operations Guide - Enhanced Maintenance and Monitoring. These messages can be customized using the language file. Bitmap images can be used in lieu of text messages. The image must end with something to the effect of "Would you like to reboot now?".

Nag messages are displayed at the top of each hour to minimize disruption during meetings and presentations.

PATCH.INI Configuration Options

All configuration options are located in the PATCHING section of the file.

ResumeMessge

This is the message that will be displayed when the workstation was off during the scheduled patch installation time and the Resume option has been selected. As in other messages, the "\n" can be used to insert line-breaks in the message.

PatchReboot

A Boolean value that controls whether the workstation can be rebooted. This controls both the pre-update and post-patch reboot. If TRUE, the workstation will be rebooted prior to application updates starting. If a user is logged in, they will receive a 5-minute grace notice.

If this setting is FALSE, then reboots will be suppressed *if a user is logged in.* If no user is logged in, the pre and post update reboots will be performed without any delay.

The following options are available only if the PatchReboot setting is FALSE *and* a user was logged in at the start of the updating process.

SchedReboot

Defines the time that a reboot will be performed if not immediately after updating. The user will receive a pop-up notice to tell them that the reboot has been scheduled. They will receive a reminder 15-minutes before the actual reboot to allow them to save their work. If SchedReboot is defined, the nagging will be not be used.

MaxNag
The number of times that a nag message will be displayed before forcibly rebooting a computer. A value of "-1" will nag forever without a forced reboot. A value of "0" (zero) will disable the display of nag messages. Any other positive value will nag the user each hour until a reboot is performed or the maximum number of nag messages have been displayed. *This value should take into account that nag messages start immediately after patching completes, which could be as early as 2 AM. To ensure that a forced reboot does not occur until 6 PM or later, this value should be 14 or higher or the MaxNagStart time should be defined to start the countdown. Use of MaxNagStart is the preferred method.*

This value can be used in conjunction with MaxNagStart, which defines when the "countdown" will begin, allowing the use of a smaller value.

MaxNagStart
A *time* value (HH:MM) that defines when the MaxNag count begins to count down. Prior to this time, the nag messages will simply state that a reboot is needed and ask if the computer can be rebooted. When the MaxNagStart time has been reached, the message changes to report the number of hours left before the computer will be rebooted.

NagAction
Defines the default action for a nag message. If set to "continue" (or undefined), the "No" button will be the default to respond to the "*Would you like to reboot now?*" message. If set to "reboot", the "Yes" button will be the default action. This will also be the default action taken if the user does not respond to the prompt within the timeout period.

NagTimeout
This defines the number of minutes that the nag message will remain visible before closing and taking the default action. The default (and maximum) timeout value is 30 minutes.

UseNagImage
This is a Boolean value (Y/N) that switches the message display from text to BMP image files. The MSP must create the image files before enabling this option! The image must be in BMP format and measure 445 pixels wide by 150 pixels high, and must be 96 dpi.

There are 3 images supported:

- PatchNag1.BMP - (NOW) Notify user of reboot in 15-minutes!
- PatchNag2.BMP - (SOON) You have a limited time before a reboot will be required!
- PatchNag3.BMP - (NEEDED) Your computer must be rebooted for updates to be installed.

When Nag Images are enabled, the count of remaining hours before a forced reboot are displayed in the Title Bar of the message when PatchNag2.bmp is displayed. When text messages are used, the count replaces the "&COUNT&" macro in the text string.

Other Operational Considerations

The **WIN-Reboot - Patching - Workstations** procedure is always run at the start of the update cycle and again when complete. This will not directly reboot the computer - the reboot is controlled by an application that reads the PATCH.INI file and acts according to the parameters defined above. Further, the VSA patch configuration is set by policy to "Do not reboot", since reboots are controlled by the app.

"Logged on" status includes remote sessions (RDP). Any connection to the computer will be detected as a user being logged on and can prevent the restart unless PatchReboot is enabled.

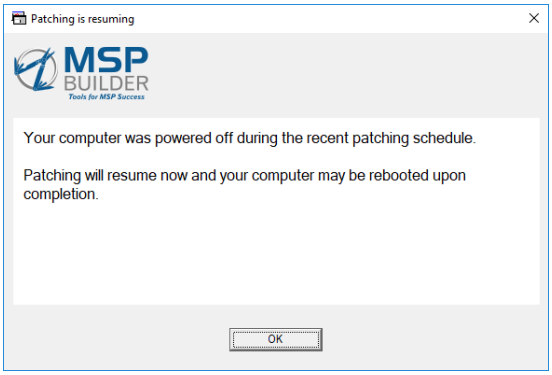
If the nag process initiates and the user later logs off, the system will reboot on the next nag cycle if the user is still logged off.

If the user manually reboots the computer, the nag process will detect the low uptime and terminate, preventing any further nagging. If the user shuts down and Fast Boot is enabled, this will not apply

patches. The nag process will recognize this and continue to nag the user until a reboot is performed once the computer is powered on and the user logs in.

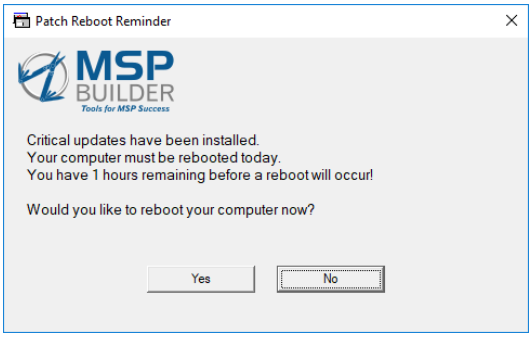
Patch Resume Notice

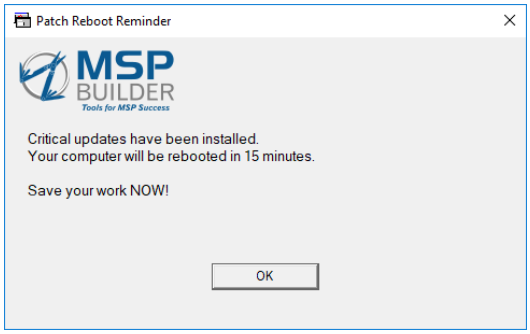
This is a sample of the Patch Resume notice that is displayed when Patch Resume is enabled and the computer was powered off overnight:



Patch Nag Messages

These are the default nag messages - the left image represents the situation where a specific number of hourly nag messages remain, the right image represents the last message where a reboot is required.





The text in these messages can be defined in the user interface Language setting file, allowing messages to be customized and automatically changed to Locale-specific languages.

Custom Patch Schedules - Warning

Successful patching depends on several processes running in specific sequences. A failure to account for any of these

Windows 10 Upgrades

Windows 10 upgrades are not considered "patches" and thus are not a part of Patch Management. MSP Builder includes an application that will perform the upgrade. Upgrades can take nearly 2 hours to complete, will impact the system performance during the upgrade, and can prevent the user from being able to use their computer at all during much of the upgrade process. Shutting down or rebooting the computer during the upgrade process can leave the computer in an unusable state.

We strongly recommend that this update process be scheduled manually only after clear communication with the client and the end-user.

Upgrade Tool Set

The upgrade process utilizes two Agent Procedures, an RMM-Suite application, and a configuration file.

- **WIN-Patching - Win-10 Platform Upgrade**
  This procedure downloads the Microsoft update tool, prepares the platform, and initiates the upgrade. Note that this procedure can report a failure if any of several conditions prevent the upgrade process, including:
  - **FAIL-CONFIG** The RMUWPU.INI file is not present. Run the **ALL-Deploy Common Files** procedure and retry the upgrade.
  - **FAIL-LICENSE** The workstation's O/S is not licensed or activated. also be reported if the MSPB Audit hasn't run.
  - **FAIL-VERSION** The workstation's ReleaseID can't be found in the Upgrade Configuration Data. Contact MSP Builder support if this happens for review and an update.
  - **FAIL-SPACE** There is not sufficient space on the computer to reliably perform the upgrade.
  - **FAIL-DLDIR** The application failed to create the Download Directory **%SYSTEMDRIVE%\Temp\W10Update**. Check permissions on the system, manually create the Temp folder if necessary and retry the upgrade.
  - **FAIL-DOWNLOAD-1** The download failed and wget.exe is not available as a backup method.
  - **FAIL-DOWNLOAD-2** The download failed both attempts. Check firewall, AV, and Internet Security settings.
  - **FAIL-AUTH** The tool was used on a system not authorized for MSP Builder tools.

  The procedure will check for Bitlocker protection on the C: drive (%SYSTEMDRIVE%) and disable protection if it is enabled prior to starting the upgrade.
- **WIN-Patching - Win-10 Platform Upgrade Check**
  This procedure runs 3 hours after the Upgrade starts. It removes the W10Update folder from the Temp directory, then checks whether the version has changed. If the update was successful, it checks whether Bitlocker protection had been disabled and enables it. It then reports one of the following status messages:
  - **NEVER - currently #** Indicates that an upgrade attempt was never initiated and reports the current Release ID.
  - **SAME - currently #** Indicates that the upgrade was attempted but the system is still at the same ReleaseID. The logs written by Microsoft's update tool should be reviewed to determine why the upgrade did not complete.
  - **UPDATED to #** Indicates that the upgrade was completed and a new Release ID was found. The new Release ID is reported.

- **RMUWPU.BMS**
  The RMM Suite utility that performs the upgrade. This tool performs the platform validation, prep, downloads the correct Microsoft upgrade tool, disables Bitlocker protection (if enabled), and initiates the upgrade.

  When invoked with the "--Check" argument, it reports whether any previous upgrade was performed with this tool, whether the upgrade was successful, and will enable Bitlocker protection if an earlier invocation of the tool had disabled it.

  This tool writes detailed messages for each action to the RMUWPU.LOG file, located in the KWorking\Logs folder.
- **RMUWPU.INI**
  This is the configuration file used by the RMUWPU tool. This file is maintained and deployed by MSP Builder and cannot be modified. The configuration file defines the currently known Windows 10 Release IDs, the disk space requirement, download URL, and the command-line arguments required to initiate the upgrade.

This Upgrade tool is not configured to run automatically. It is the responsibility of the MSP to notify users of the upgrade schedule and then schedule the upgrade manually. There is very little load on the VSA, so several upgrades can be scheduled at a customer location simultaneously. As each computer will download the upgrade files directly from Microsoft, consideration should be given to staggering the schedules to minimize the effect on Internet bandwidth. 50 agent upgrades per hour would be a reasonable target when run outside of normal business hours.

After the upgrade completes, there will be several folders in the root of the primary drive. These can be removed once the upgrade has been validated. This task is left to the discretion of the MSP to determine if and when these should be removed. The commands below for the cleanup can be placed into a batch file, deployed, and executed if desired, or if space reclamation is required.

```
@Echo Off
REM - Remove the Windows.old folder
takeown /F %SYSTEMDRIVE%\\Windows.old\* /R /A
cacls %SYSTEMDRIVE%\\Windows.old\*.* /T /grant administrators:F
rmdir /S /Q %SYSTEMDRIVE%\\Windows.old\

REM - Remove the $GetCurrent folder
RD /S /Q "%SYSTEMDRIVE%\\$GetCurrent"

REM - Uninstall the W10Upgrader app
"C:\Windows10Upgrade\Windows10UpgraderApp.exe" /Uninstall
RD /S /Q "%SYSTEMDRIVE%\Windows10Upgrade"
```

This script example is provided "as-is". MSP Builder is not responsible for the accuracy or safe execution of this process on client systems. We recommend that you verify the current commands and test before deploying this to a production environment.

System Policy Management

System Policies are an integral part of Core Automation and the RMM Suite's design. Their use allows a high degree of consistency to be obtained in the configuration of the agent, deployment of monitors and scheduled tasks, patch procedures, and even remediation of specific conditions. The extra initial effort to define a few customer-specific policies during onboarding is far outweighed by the gains in simplified administration.

Overview

Policies define a specific set of configuration parameters. This might be, for example, all of the settings and tasks needed to patch a computer. In this case, the patching policy would perform the following steps:

- Define membership in Patch Policy groups
- Define the source for patches, such as Internet or a local cache folder
- The reboot policy
- Conditions under which certain failures generate a ticket or other notification
- Configure Windows Update
- Define a schedule for patch scans
- Send an email notification before patching
- Run a procedure to disable alerts before patching
- Run a procedure to reboot the computer prior to patching
- Invoke the patch install process on a specific schedule

You can see from the steps above how a single policy can simplify administration (and reduce errors) compared to performing each of these steps individually. For workstations, all of the settings are usually defined in a single policy, linked to the Global Root (Auto-Pilot) or the customer's Workstations machine group (customer-specific policy). Servers utilize two policies – one to define the common settings and scan schedule and a second to disable alerts and perform the scheduled pre-patch reboot and the actual patch install. The schedule policy is defined using the Auto-Pilot policy that applies based on the Patch Schedule custom field, allowing independent scheduling of every server. The MSP Builder Server Patch Management tool allows configuring server-class patch schedules using an Excel spreadsheet.

The Patch Schedule custom field is used by servers to allow anyone with Audit Machine - Edit capabilities to define the schedule. The standard patch schedules use the following format:

- None — No patching is performed on this server.
- Manual — Only manual patching is performed on this server.
- Pxxx — Patching is performed by the defined schedule using Patch Management.
- Sxxx — Patching is performed by the defined schedule using Software Management. *Software Management Patching is not currently supported.*

For both Patch Management and Software Management, the "xxx" represents a schedule code as follows:

First Character	Process, P=Patch Management, S=Software Management
Second Character	Week of deployment, 3, 4, or 1 (cycle starts on week 3)
Third Character	Day of deployment, A=Saturday, U=Sunday, W=Wednesday
Fourth Character	Window, 1-9, a-w. 30-minute increments starting from 00:30 Sat or 07:30 Sun.

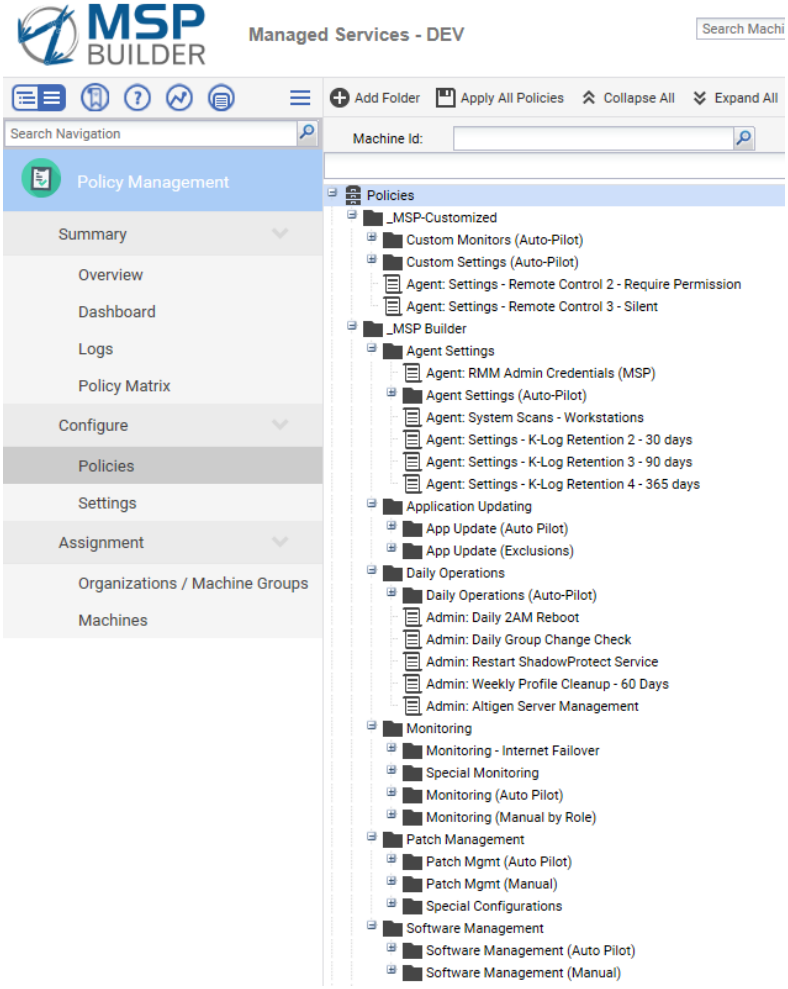
Thus, the code "P4Uc" would be "Patch Management, updates at 4th Sunday at 08:30". Note that the time indicated is when the server will reboot, actual patching will start 30-minutes after that.

The use of our Offline Server Patching tool is recommended to simplify administration of server patching.

The MSP is free to expand this concept and define their own schedule by creating policies and views using the provided components as an example. Note that custom configurations will not be supported.

Standard RMM Policies

More than 225 standard policies are included with the RMM Suite. While most can be used without modification, some tuning may be needed to address customer or MSP requirements. The Agent Setting and K-Log Retention policies primarily contain MSP-specific settings and should be customized during installation of the Core Automation components. These are located in the _MSP-Customized folder.



Policies are general in nature and can be applied to any customer. Policies that have an "(Auto)" designation are "Auto-Pilot" policies, linked to the Org Root and their application is controlled automatically by Views. Auto-Pilot policies are not customer-specific. The Auto-Pilot policies will be discussed in detail later in this section.

Clients

A structure to organize customer-specific policies should be created when such policies are needed. A "Clients" folder should be created, with subfolders to hold customer-specific policies named by client. These should have a "(Cust_ID)" designation to quickly identify the customer that the policy was created for. Creating the subfolders and these policies is the responsibility of the MSP that uses RMM Suite.

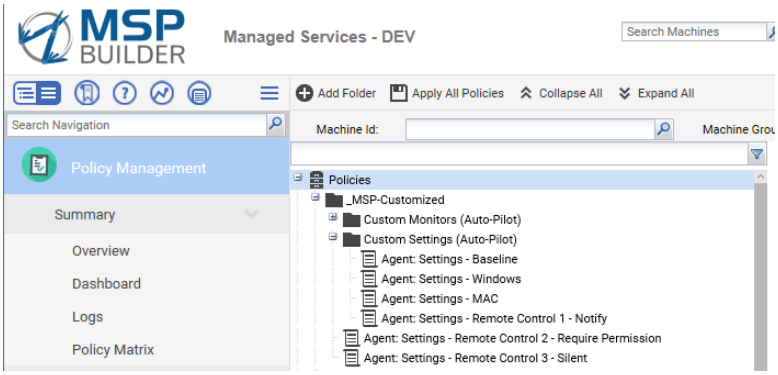
Note: We generally recommend *against* creating client-specific policies! Most of these are either exceptions to the standard operation or additional features. Since it is rare for one client to have such exclusive requirements, it is much better to create a generic policy that either *overrides* the core policy settings or *adds* the additional settings not provided in the core policies. These can then be linked to multiple clients or even added to the Org-Root and their deployment controlled by views. Custom monitors, for example, would be linked globally and applied where detected rather than linking and being applied to all client systems. This reduces platform administration while increasing accuracy and consistency in deployment.

MSP-Customized Policies

The policies contained in the "_MSP-Customized" folder are designed to be applied to any customer after customization by the MSP. The subfolders noted as "Auto Pilot" are linked directly to the Org Root.

Custom Setting Policies

The core policies perform global configuration tasks and are usually linked to the root of the Machine Group organizational structure. These are also the policies that need the most attention after installing the RMM Suite to define the MSP-specific settings.



- Agent: Settings - Baseline
  This performs most of the agent configuration and requires the most customization. It defines settings for the Agent Menu, including MSP company identification, portal URL, contact information, Remote Control notification settings, and Agent Check-In settings. The View allows this to be applied to all agents.
- Agent: Settings - Mac &
  Agent: Settings - Windows
  The Kaseya Working Directory is defined in this policy and is unique to Mac and Windows platforms. The View restricts the policy to either Windows or Mac platforms.
- Agent: Settings – Remote Control *
  These 3 policies should be updated with custom notification messages. The Require Permission and Silent policies are usually applied per-customer only where required.

MSP Builder Policies

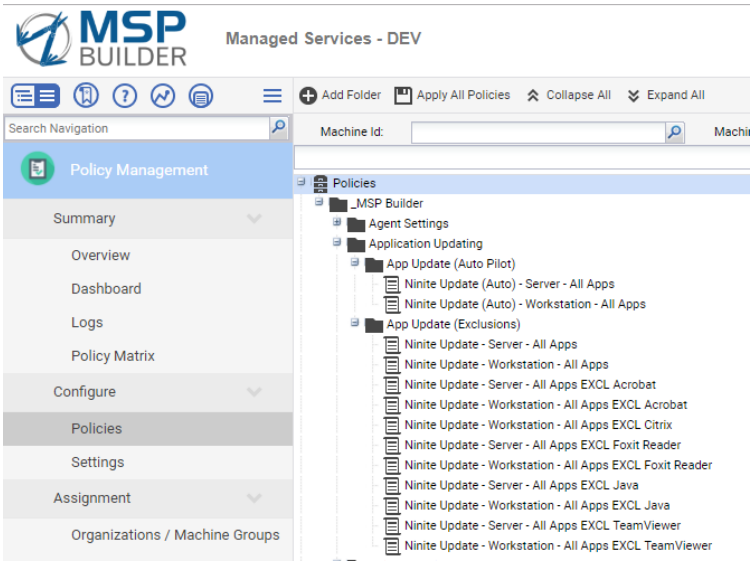
The policies contained in the "_MSP Builder" folder are designed to be applied to any customer. This is a large collection of policies and cover all the typical MSP needs for agent configuration and VSA automation. In most cases, the subfolders noted as "Auto Pilot" are linked directly to the Org Root.

- Agent: Settings - K-Log Retention <type>
  Several policies that define the Kaseya Log Retention settings. The default policy has settings recommended by Kaseya that balance logging with storage consumption. Other policies extend he default settings and can be applied to specific customer folders if extended logging is required. The remote-control logs are retained for 365 days in all of these policies.
- Agent: Upgrade - MAC / Windows
  This will cause the Kaseya Agent upgrade to be initiated when the detected version is below the defined level. The agent is usually upgraded manually on a few systems to verify operation, then the View associated with these policies is updated with the current version number. Upgrades occur automatically within a few hours.
- Agent: Audit - Monthly Server / Workstation
  These two policies perform the same Audit – Full System procedure but use separate schedules for servers (first 3 days of the month) and workstations (second week of the month). This audit augments the standard Kaseya audit by collecting additional configuration information and populating custom fields or uploading configuration files. The views restrict these policies to Windows platforms, either servers or workstations.

- Agent: System Scans - Servers / Workstations – schedules a regular scan on all agents. This is a Kaseya scan that identifies new services and other system components. By default, servers run monthly, and workstations run quarterly.

Application Update Policies

These policies utilize NinitePro or Chocolaty to perform regularly scheduled application updates. Each policy includes a configuration and a schedule, so these should be reviewed, and schedules adjusted to local requirements if necessary. Each update process has two policies – one for servers and one for workstations. The configuration settings are similar but the schedules are unique – weekly for workstations and monthly for servers.

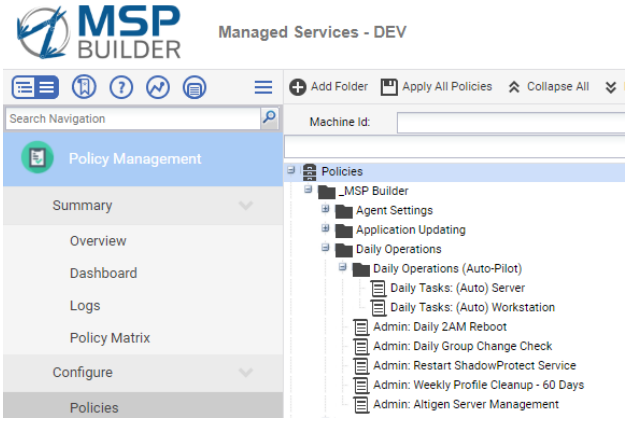


The primary set of policies performs unrestricted updates – any application found that is supported will be updated if a newer version is available. The remaining policies define a single application exclusion. If a customer requires multiple applications to be excluded from regular updating, a new Update procedure will need to be created along with its corresponding configuration file. A policy set can then be duplicated and modified to invoke the new update procedure.

Auto-Pilot policies (identified with "(Auto)" are provided with full-updates scheduled using the default schedule. Server updates are scheduled for the Second Saturday of each month, a day when typically, no patches are scheduled for installation. Workstation updates are scheduled for the evening prior to weekly patch installations to minimize the number of system restarts that are needed. The policy that applies all updates runs over a 2-hour window starting at 03:30. Several policies are provided that exclude certain applications - these policies run exactly at 03:30. When custom application update policies are applied, the "APPUD" Blocker ID should be applied to disable the Auto-Pilot policies.

Daily Operation Policies

The policies in this folder apply common tasks that run on a daily basis. These include many common system-administration tasks.



- Daily Tasks: (Auto) <type>
  This policy schedules tasks that include clearing the daily audit data file, initiating the smart monitors, optionally invoking daily maintenance tasks, and then running the daily audit. The Maintenance procedure is invoked if the MAINT policy blocker is not defined.
- Admin: Daily 2AM Reboot
  Used to reboot workstations every day. This policy is restricted to Workstations (any type) and provides a 10-minute grace warning prior to the reboot.

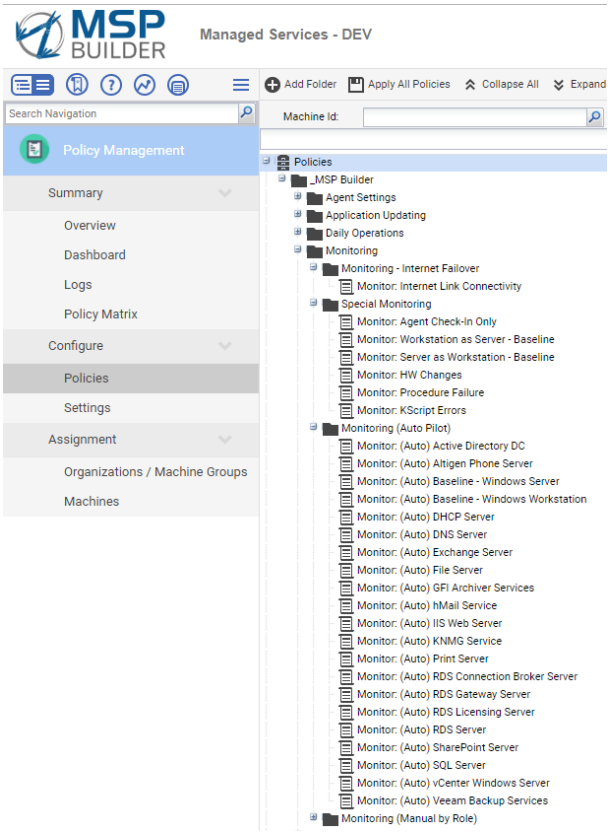
- Admin: Weekly Profile Cleanup – 60 Days
  This policy schedules a user profile cleanup procedure that removes any profile not used within the prior 60 days. This is generally linked directly to RDS type systems, and the view restricts it to Windows Server class systems. The cleanup is invoked weekly between 20:00 and 21:00 each Friday. No reboot is required, but this process should not be scheduled concurrent with patching or other tasks that could reboot the system.
- Admin: Restart ShadowProtect Service
  As the name implies, restarts the ShadowProtect service to clear memory and cache, improving the health of certain Shadow Protect platforms. This policy is linked directly to servers as it is only needed in specific situations.
- Admin: Altigen Server Management
  Performs regular maintenance tasks on Altigen VoIP Phone Servers.

Monitor Policies

These policies are used to assign monitor sets to machine groups or specific machines. There are two "baseline" policies that contain the common monitor sets for servers and workstations, and these are Auto-Pilot policies linked to the root. These can be disabled on a per-agent basis by applying the BLMON Blocker ID.

The Special Monitoring folder contains policies that are linked to individual servers or machine groups for special purposes. There are two special policies that apply Workstation monitors to Server platforms and vice-versa. The Agent Check-in Only policy has no other monitor besides an agent check-in. The monitors for Procedure Failure and KScript Errors should be applied only at the direction of the MSP Builder support team.

The remaining policies apply application-specific monitor sets and are also the Auto-Pilot type. These have a specific view defined that applies the monitor based on platform type (workstation or server), and System Role ID. These role-based policies can be blocked altogether by applying the "EXMON" Blocker ID, or a role-specific Blocker ID.



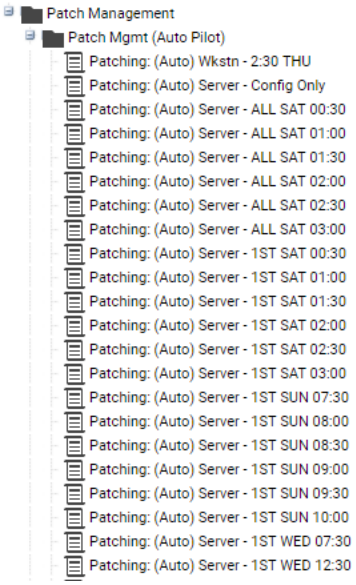
Of special note is the Monitor: Internet Link Connectivity policy. This policy will automatically install the link connectivity monitor tools, define the monitor set, and then schedule the hourly checks. This policy should be linked to a specific computer to monitor the primary/backup Internet links. The policy should be linked the first time when the primary link is active or manual configuration will be required. See the man-page for this utility for full configuration and operational information.

Note that the policies in the "Monitoring (Manual by Role)" folder are identical to the Auto-Pilot policies but are provided for manual linking to systems (usually in the Special group) to accommodate special situations.

Patching Policies

There are three general types of policies used for patching.

1. The policy defines all configuration settings AND installs the updates on a defined schedule. This type is primarily applied to workstations as they rarely have restrictive update schedules. It is generally linked to the customer's workstation machine group. There is an Auto-Pilot policy that applies the default patching configuration and schedule to workstations, and this can be blocked by applying the "PATCH" Blocker ID.
2. The policy defines all configuration settings but does not install any updates. There are two standard policies of this type and are applied to server class systems. One applies to servers and the other is used to update workstations used as servers. These policies are often copied and modified to contain customer-specific settings, such as membership in additional patch exclusion polices. When customized, this policy type is usually linked to the customer's server machine group. There is an Auto-Pilot policy that defines the default configuration settings for servers. All server configuration policies require the next type of policy to schedule the outage window and perform the update installation.
3. The final policy type is used to deploy the updates based on a specific schedule. This policy invokes two procedures – one to suspend alerts during the patch window and one to reboot the computer 30 minutes prior to the patch installation process. It then schedules the patch installation. These policies are linked directly to individual computers to define a specific installation time by mapping the value in the Patch Schedule custom field. Core Automation provides the following schedules, and these can easily be customized, duplicated, and expanded to suit MSP and customer needs:
    a. Every Saturday, 00:30-03:00 on 30-minute intervals
    b. Third Saturday, 00:30-04:30 on 30-minute intervals
    c. Third Sunday, 07:30-11:30 on 30-minute intervals
    d. Third Wednesday, 07:30 and 12:30
    e. Fourth Saturday, 00:30-04:30 on 30-minute intervals
    f. Fourth Sunday, 07:30-11:30 on 30-minute intervals
    g. Fourth Wednesday, 07:30 and 12:30
    h. First Saturday, 00:30-04:30 on 30-minute intervals
    i. First Sunday, 07:30-11:30 on 30-minute intervals
    j. First Wednesday, 07:30 and 12:30



Example patching policies – Config and Schedule for servers and combined policy for Workstations

Limitations:

Windows patching and application updating require careful planning and scheduling, which has been taken into account by our updating solution.

- MSP Builder does not provide support for customer-defined patch schedules.
- Custom schedules defined on Monday or Tuesday will likely not operate as expected, being too close to the patch scan schedule.
- Creating alternate Patch Scan schedules can result in Windows Update randomly applying updates and rebooting computers. Multiple patch scan schedules must be avoided.

Using Policy Templates

The policies can be used to create customer-specific policies with minimal effort.

- Select a policy similar to the one you want to create
- Copy the current Name, click Save As, then paste the copied name. Adjust the name to include the CustomerID. Click Save. The new policy must be moved to the MSP folder to avoid loss during platform updates.
- Expand the Clients folder, the alphabetic subfolder, and the customer folder (create it if necessary).
- Select the new policy and drag it to the customer folder.
  Hint! Hold the policy over the folder until you see the destination folder "shake", then drop it! Failure to do so can cause the policy to be placed into an inaccessible location.
- Edit the policy configuration as necessary. Be sure to create and then select the appropriate client-specific view.

System Configuration Notes

There are two system configuration parameters that must be adjusted. During the initial deployment, the Deployment Interval should be relatively short – 5-10 minutes. Within a week of getting most policies deployed, this value should be reduced to 30-60 minutes.

The compliance check verifies that policies are being applied properly. This configuration should generally be performed on a 4 times daily schedule, starting at midnight, and running every 6 hours (360 minutes). The process should be scheduled to be distributed over the entire period.

Note that when diagnosing compliance issues, the compliance check can be set to a very short value. This low value should be used for as short a time as possible as it increases the load on the VSA platform.

Defining Client-Specific Policy Views

When customer-specific policies are used, it is important that they be applied only to the intended client. Views provide the ability to restrict the application of policies to specific customers and computer types. Core Automation uses three Policy Control view prefixes:

- XAPC Auto-Pilot Control
- XARC Auto-Remediation Control
- XPOL General Policy controls

When the MSP needs to create customer-specific view types, they should utilize a prefix of "ZCPC-", which places them well after the common views used to filter agent reports. These represent a "Customer Policy Control" name prefix. The defined name structure for these views is "ZCPC-*customer-type*". It is the responsibility of the MSP to create client-specific views.

Important Notes Regarding MSPB Policies!

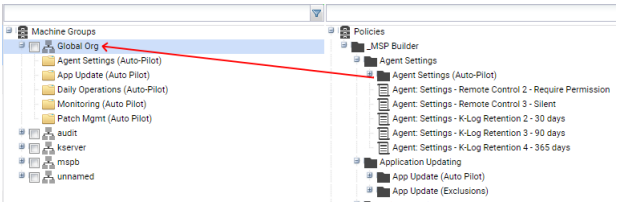
Do not create policies inside the _MSP Builder folder structure – always use the _MSP-Customized folder structure for custom policies.

Do not intermix custom policy control views with MSP Builder's policy control views. These could be lost during an RMM Suite upgrade. Always create a copy of our views for use with custom policies.

Always distinguish between the core RMM Suite components and any that you customize, either by using an alternate name prefix (such as YAPC_) or a unique folder location!

Assigning Policies

Policies are assigned to agent computers by either dragging the policy onto the machine-group folder or by adding a policy to a specific machine through the add policy option. Using the drag and drop method allows policies to be rapidly applied to all agents or a specific group of agents. Assigning policies directly to an agent computer allows specific control. Note that the default configuration allows the folders marked "(Auto Pilot)" to be dragged directly onto the Global Org folder, as shown here.



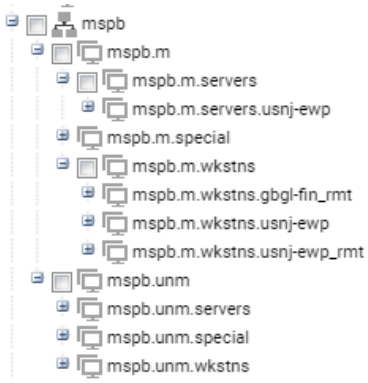
Globally

Some policies will have settings that apply to every agent. These are referred to as "global" policies and are linked to the "Global Org" group. All of the Auto-Pilot policies are linked here.

By Client & Machine Group

This method allows policies to be applied to specific customers and groups, which can override any Auto-Pilot policy settings. These policies may be the "(Common)" type that have custom settings but are not customer-specific, or policies defined specifically for the customer, such as the antivirus or antimalware profile definitions.

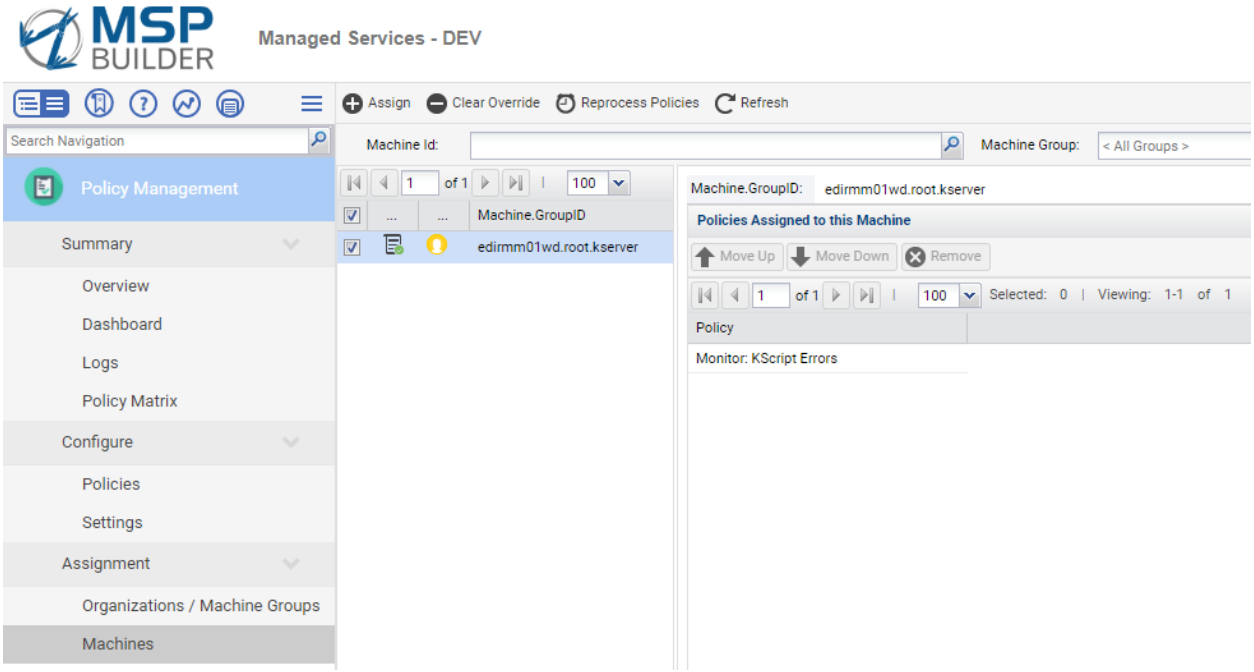
The most common way to apply policies is to link them to each customer's machine group. Refer to the previous illustration to understand how policies are linked to groups. Starting with the "m" group, policies that apply to all agent types are linked here. This underscores the importance of the "m" or root machine group discussed earlier for managed clients. The policies linked here typically configure overrides for a specific customer. The policy should define settings for both servers and workstations. It is linked here and will apply to machines in all lower groups. This is a good example of a customer-specific policy.



The "wkstns" group has been expanded in this illustration, showing the location sub-groups. Policies can be applied to the workstation level and apply to all locations, or to a specific location. This illustrates the flexibility of the RMM Suite's organization machine-group hierarchy.

By Specific Machine(s)

Applying policies to specific machines provides a high degree of control. The image here shows a server that has one policy specifically linked to it. This policy configures this server to alert when KScript failures are detected. Note that this is an example of associating a specific policy with a machine. SQL and Exchange are typical of additional monitor sets that are applied to individual servers, especially to monitor database-specific objects.



When linking policies to machines, make sure that the compliance filter is turned off (see below). If the filter is enabled, only systems meeting the filter criteria will be displayed.

Auto-Pilot Policies

These policies serve two critical functions within Core Automation:

- By automatically applying policies for commonly performed tasks using widely supported default values, they simplify the deployment of
  - Baseline Server and Workstation monitor sets
  - Extended, Role-Specific monitor sets for Servers
  - Regular, daily tasks like the initiation of Smart Monitors and maintenance procedures
  - Default patch configuration (Servers & Workstations) and their update schedules
  - Default application update schedules for Servers and Workstations
- For servers in particular, the automatic application of role-specific monitors ensures that critical alerts are not missed, as they might be when manually applied monitors are depended upon. Since these monitors are applied dynamically, changes to servers – adding or removing applications or roles – will automatically add or remove the corresponding monitor set within 24 hours. If faster application (or removal) of monitor sets is desired, simply run the Daily Audit procedure against the server and the Auto-Pilot monitoring changes will be made within a few minutes of when the audit procedure completes.

An Auto-Pilot policy is a standard policy that defines monitors, procedures, basically anything that you want to define. What sets it apart from a typical policy is that the View assigned to the policy defines the conditions under which the policy should be applied. There are typically two, three or four criteria in an Auto-Pilot view:

1. The view is restricted to agents that are not suspended. The remaining filters use Advance Data Filtering to control the application.
2. The view defines the exact operating system versions where it may be applied. This parameter will need to be updated when new platforms become supported or old platforms reach end-of-life status. Using OS Version numbers allows very specific control and selection of specifically supported platforms.
3. A Role ID can be matched in the "System Roles" custom field. This field is updated at least daily from the "RMM/System Audit/Data Query" procedure on every managed agent.
4. When System Roles are used to apply a policy, the "Policy Control" custom field can be evaluated to restrict the policy. This field holds "Blocker IDs" that prevent the policy from applying. Each Blocker ID is surrounded with a dash (-) character. All Auto-Pilot policies have a corresponding Blocker ID as listed below. The Extended Monitor type of Auto-Pilot policy provides an additional policy-specific Blocker ID.

This can best be illustrated with an example. Assume that the agent has SQL Server and IIS installed. It's System Roles field will contain "- SQL IIS". (This field is always prefixed with a dash.) If neither of these services need to be monitored, the "EXMON" blocker would be applied, and the "Policy Control" field will contain "-EXMON-". This prevents ALL Auto-Pilot policies that apply extended monitoring from being applied. If it was desirable to monitor SQL but not IIS, then the IIS blocker would be defined. This would prevent IIS but not SQL monitors from being applied. In this case, the "Policy Control" field would contain "-IIS-" only.

Standard Blocker IDs

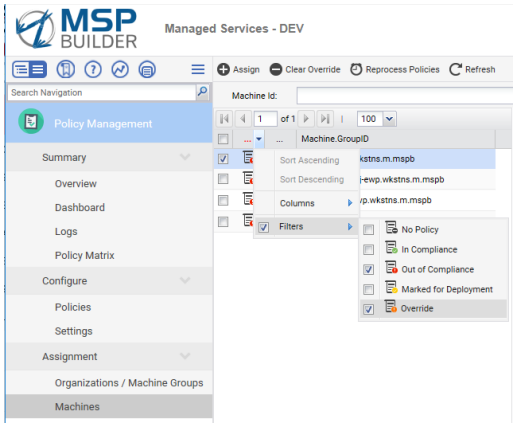
There are currently 5 standard Policy Blocker IDs defined that broadly control the application of Auto-Pilot policies. Each is applied or removed with an Agent Procedure, which also maintains the state of the Policy Blockers in the registry on the agent system.

- BLMON Prevents the Baseline monitor set from being applied.
- EXMON Prevents *ALL* Extended monitor sets from being applied.
- MAINT Prevents the standard maintenance tasks from being run. This is applied when the customer has requested that maintenance tasks not be run.
- APPUD Prevents the standard Application Update procedures from being run. This blocker is used when a custom application update policy is applied, or application updates are not run via automation.
- PATCH Prevents the default Patch policy from being applied. This is used when custom patch configurations, especially additional patch block groups, are to be applied to the customer's computers, or when patching is not provided through Kaseya.
- <Role_ID> A specific role ID is used to block specific Extended monitor sets from being applied. The Role ID must match the Role ID defined in the "System Roles" field and must be surrounded with dashes (-). Like all Blocker IDs, the agent procedure should be used to ensure that the list of Blocker IDs is synchronized with the local copy on the agent. These are added and removed by using the Policy Management procedures that allow you to specify a Role ID.

The System Roles and Policy Control fields should not be updated manually through the Audit / System Information page, even though this method is possible. The detected roles are updated daily via the audit process, and the Policy Control blocking IDs are maintained on the local system. Editing the custom field data will cause the data to become out of sync with the computer. When a procedure is run to add or remove blocking IDs, the work is performed using the agent copy of the data, which is then used to update the custom field. It is recommended that the procedures always be used to manage the Policy Control field data.

Maintaining System Policies

System Policies have specific maintenance tasks that are performed on a regular schedule. When new policies are created, these maintenance tasks might be performed daily or even more frequently. Once policies are defined and tested, the maintenance tasks can be performed less frequently – generally once per week.



When checking for compliance or overrides as part of regular maintenance, it will be helpful to turn on the compliance filters as shown here in the image.

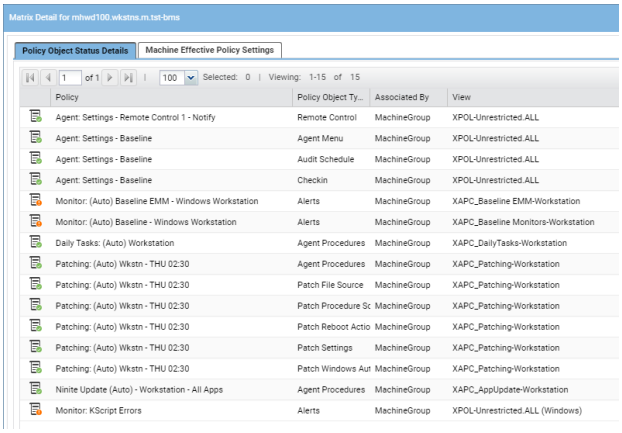
Checking for Compliance

When policies are deployed, there may be certain conditions that prevent a setting from activating. This condition is detected, and non-compliant systems are represented with a red icon.

A common reason for non-compliance is when a policy contains settings that only work on a Windows platform but is applied to a Mac or Linux machine. The solution in this case is to change the view associated with the policy (or possibly create a new view). Core Automation standard policies have been created with appropriate views that restrict by type (server or workstation) as well as platform (Windows or Mac).

Other situations that can cause a non-compliant condition include local settings that conflict with the policy settings. These conditions require more investigation.

When a non-compliant system is identified, the reason for the non-compliance can be discovered by hovering over the red icon. The impacted policy sections are identified by individual red or orange icons. Note that in this case, all of the compliance issues are related to overrides - either a manual setting being applied that the policy cannot change, or a manual change applied after the policy was applied.



Policies can be easily forced back into compliance. Start by filtering for Overrides. Select the displayed agents and click the Clear Override menu option. Overrides are also compliance issues, so if you search for compliance failures first, you will miss the overrides! Once the overrides are resolved, change your filter to Non-Compliant, select the agents, and click the Reprocess Policies menu option. After a minute or two, click Refresh - the display will be blank when all compliance issues are resolved. If many agents are non-compliant, it could take several minutes for them to return to a compliant state.

This was an example of a typical issue, but each situation is unique and troubleshooting system policy compliance takes a bit of detective work. One challenge to be aware of when troubleshooting is that a bad setting in one policy could cause related settings in other policies to show as non-compliant.

Managing Overrides

Overrides can occur when a VSA setting, including linking of monitor sets, is applied directly to an Agent instead of through a system policy. This is most often the result of old habits – an administrator needs to monitor a new event log or service and simply adds it directly.

While it is possible to simply select the agent from the Policy Management / Assignment / Machines menu and click “Clear Override”, this will remove the manual setting from the agent and prevent the

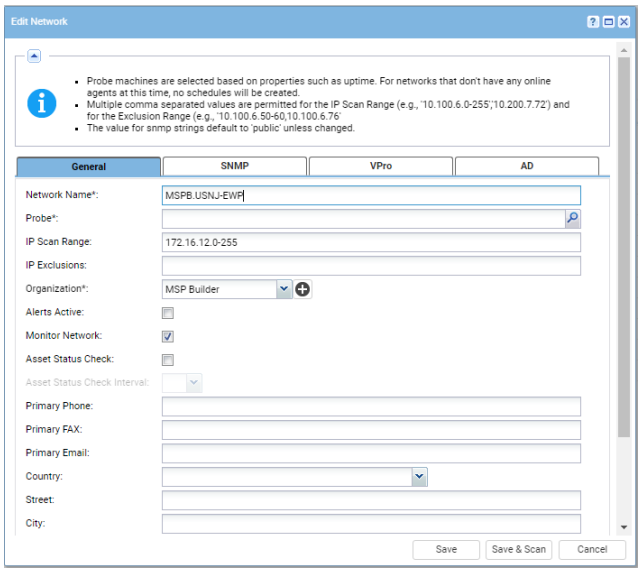
setting or monitor from doing its job. The proper approach is to determine what the overriding settings are, then create and apply a new System Policy to do the job correctly. Once the new policy is defined, the override should be cleared first before linking the new policy to ensure that any manual configuration is properly removed. After clearing overrides, click on the Reprocess Policies link to ensure that all updates are pushed to all affected agents.

Network Monitor

Network Monitor is used within Core Automation for two core types of monitors:

- Operational Availability – these include monitors that communicate with systems and expect specific responses. While an agent-based monitor can determine if a system service is running, the KNM Operational Availability monitor can determine if the service is actually responding properly to requests. These monitors also perform general server availability checks by verifying that the server responds to a specific status query.
- System Performance checks – All performance monitoring is done within KNM, which provides several advantages over agent based checks. These benefits include a GUI graph of performance levels over the past day (or – optionally, several days); the ability to quickly deploy performance tiers appropriate to the system capabilities; and the ability to suppress alarms during specific periods, such as when backups or other resource-intensive operations are occurring.

Integration with Discovery



Network Monitor is an agentless system and uses Out Of Band (OOB) monitoring methods. Discovery is used to map a network, and the discovered devices appear in the KNM network object. Monitor sets are applied to each device and are monitored via a "gateway" system. The gateway platform must have an agent installed.

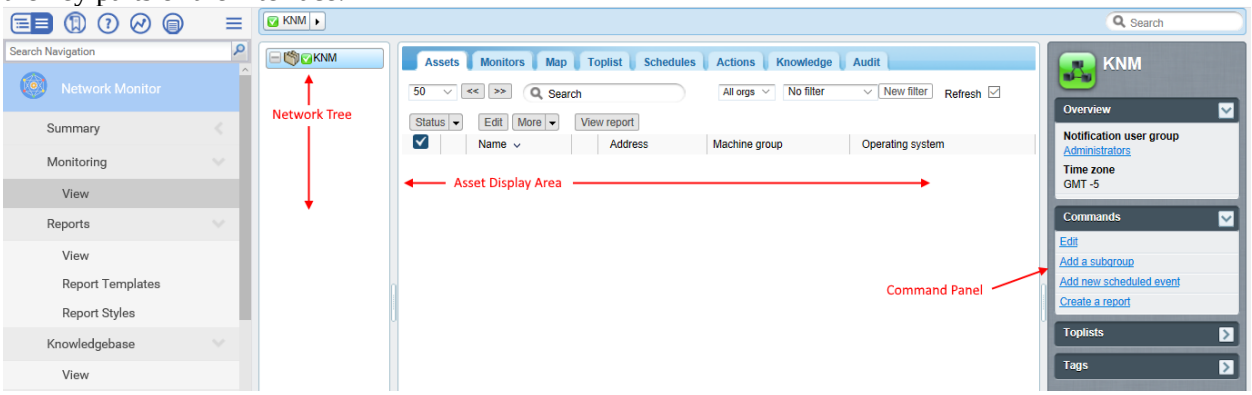
The name of the network defined in discovery is the name that will appear in KNM. The network name should be chosen to identify the customer, location, and optionally the subnet or VLAN. We require the format "<CustomerID>.<SiteID>" be used for as this can be parsed during ticket intake to identify the customer and the location. When you monitor individual VLANs at a customer, you can add a ".VLAN_Name" designation to the network name.

When discovery creates a network, it will appear in the root of the KNM structure. Do not create groups to hold related networks as this will mask offline gateway notifications! Using the Customer.Site network naming format will keep all customer networks together.

The image to the right shows the typical configuration settings for a new network – defining the network name, deployment probe, the associated organization, that the settings for Alerts Active (off) and Monitor Network (on). Unless you wish to scan a specific range of IP addresses or multiple ranges, the IP Scan Range should be left blank. We recommend a separate network be defined and scanned for each subnet or VLAN. This takes longer but provides better information as to where the issue is occurring.

Preparing for KNM

Understanding the KNM interface is essential to being able to manage it effectively. This image shows the key parts of the interface:



An important step is to define the account that will be used to access the computers. Select KNM from the Network Tree, then click Edit in the Command Panel. From the Edit Group panel, complete the following tasks:

- On the Basic Properties tab:
  - Enter a description (optional)
  - Set the Alarm Subject to:
    `KNM|Info|%[group.name].%[asset.name]|PERF-%[monitor.name]||Perf.CTR.S.P5.Alm`
- On the Advanced tab:
  - Set the Google Map Display to "Gateway"
  - Enter the Geographic Location data – street address, then city, state, country (USA)
  - Verify the Timezone setting is correct
- On the Authentication Tab:
  - Define Windows domain credentials:
    Domain or computer = LEAVE BLANK
    Username & Password = local credentials available on each server

Click the Save button.

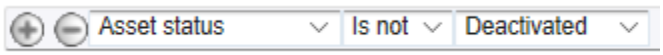
Be sure to define the credentials above in the NM_UserID and NM_Password Managed Variables. This account will be created on every agent in a server group when the Agent Init procedure runs.

Before proceeding, make sure you downloaded and extracted the KNM_Templates from the MSP Builder website to a local folder.

- Navigate to the Settings / Device Templates section.
- Click Import, Browse, and navigate to the folder where you extracted the KNM monitor templates. Select the desired template and click OK, then click Import, then OK.
- Repeat the above step for each KNM Monitor Template.

Create an Active Devices filter:

- Select KNM from the Network Tree and select the Assets tab.
- Click the New Filter button and set the filter attributes as shown here



- Save the filter as "Active Devices"

Deploying a Gateway Service

Kaseya Network Monitor needs at least one system in each monitored subnet with an agent installed. The Gateway Service is installed on this host, which performs the OOB monitors against the discovered devices. The gateway can monitor computers, printers, routers, switches, and other devices that support WMI or SNMP protocols.

Selecting a Host

There are two criteria for selecting a host for the gateway service:

- The system should be highly available, usually a server that is on all the time. A domain controller or file server are usually excellent choices.
- The system should have sufficient resources available to run the service and make the queries to the other computers and network devices. The gateway does not place a heavy load on the system, but systems that are already heavily loaded should be avoided. These could prevent consistent communication with the monitored devices, which could result in an unwanted alert.

Once you have decided on the Gateway Host, select the network from the Network Tree panel, then click the Install Gateway link from the Command Panel. Select the agent from the drop-down list and click Save. In a few minutes, the new network icon will change from blue to green, indicating that the Gateway service is checking in and ready for work. The Overview area of the Command Panel will then display information about the network, including the hostname of where the Gateway is installed.

Once the gateway checks in, you should immediately configure the gateway device to disable WMI queries. WMI won't function on the gateway machine for KNM queries (it does NOT affect WMI itself).

- Click the gateway system – the Overview in the Command Panel should display that agent's information.
- Click Edit in the Command Panel and select the Advanced tab.
- Deselect the "Use WMI" checkbox, then click Save.

The last step is to hide the workstations and other devices you don't need to monitor.

- Select the network from the Network Tree panel.
- In the Agent Display area, select each system that you do not want to monitor. These are usually workstations, but may also include printers, storage devices, and other non-enterprise assetss.
- Click the Status drop-down and select Deactivate and confirm your choice. If you have no filter set, the assets will turn gray. If the Active Devices filter is selected, these assets will be hidden, leaving only the servers and other devices of interest.

Other Configuration Tasks

Most of the configuration settings are inherited from the root of the monitoring structure. Where clients have special requirements, the following settings should be reviewed and adjusted where necessary. These can be applied to the individual networks as needed. The configuration parameters can be changed by selecting the network folder and clicking the Edit option in the Command Panel.

- Alternate notification groups – used to send KNM alerts to additional or customer-specific staff;
- Custom alarm messages (rare);
- A time zone different from the KNM site;
- Define contact information, especially if not easily accessible elsewhere;
- Set customer-specific credentials for Windows, SSH/Telet, SNMP, and VMware.
- Geographic Location, which controls the map display. This change is fairly common!

Applying Monitor Templates

Monitor Templates allow standard monitor sets to be deployed with a minimum of manual effort. The RMM Suite provides a large collection of templates for monitors and alerts. In Core Automation, the templates that begin with "_Monitor" are configured to monitor specific settings without generating alerts. These allow an engineer to review the state of the system, especially when planning to enable performance alerting. Templates that begin with "_Alert" are configured to generate an alert when the condition has crossed the threshold for a sustained period.

Monitor Settings

The monitors perform a test on 60-second intervals. The thresholds are set such that they would never trigger under normal conditions. They also require that the threshold is crossed 999 times. The Monitor Set ID type is "INF", which causes any alert that makes it past these restrictions to be discarded by the Intelligent Ticket Processing service. A monitor might meet the requirements to generate an alert if the authentication credentials change and the credentials aren't updated correctly.

Alert Settings

Monitors that can generate an alert are usually checked on a 2-minute interval. The threshold is set, particularly for performance monitors, at a level based on the hardware capabilities. Three "tiers" of performance monitors are provided in Core Automation. The threshold must be exceeded for 15 times before an alert is triggered, requiring that the threshold be exceeded for a total of 30 minutes. This prevents most spurious events from generating an alert. Once an alarm is triggered, the monitor cycle is increased to 5 minutes.

Operational Availability monitors have a slightly different configuration from Performance monitors. The check cycle is reduced to 60 seconds, which will trigger an alarm after crossing the threshold 15 times. This generates the alert within 15 minutes, although this can be reduced where necessary. This provides a reasonable balance between notification and false alerts when a system is restarted, or other transient condition occurs.

Note that Core Automation provides two types of template – server based and application based. In application-based templates, a collection of individual monitors needed by a specific application are combined to create the template. These could be a combination of Monitor and Alert types. Server templates are a collection of application templates as well as individual monitors that are applied to a specific type of server, such as a Domain Controller, RDS host, SQL server, etc.

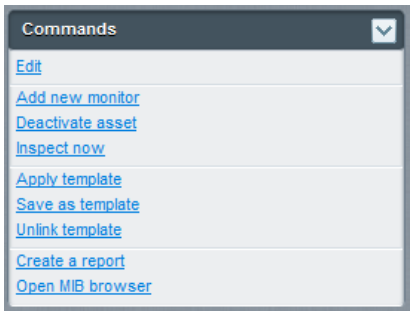
Application templates allow you to quickly add a set of monitors to a server and are useful when a new service is installed or a specific application needs to be combined with a Server template. Server templates allow the most common collections to be applied to a server, and this is the most common way of assigning monitors within KNM. If a specific set of monitors are often applied, creating new Server templates would be an effective solution.

Naming Templates

The MSP Builder method for naming templates places the custom and most-used templates near the top of the list. Server Templates have a prefix of two underscore characters, where other custom templates are prefixed with a single underscore. Since these special characters appear early in the sort order, MSPs are encouraged to use a different special character to identify their customized templates. The "~" and "!" are good choices for this purpose.

Customizing the Monitor Sets

The provided monitor sets are suitable for a wide range of application environments. There are times, however, where customization may be required. Starting with a template and then customizing it can save time over creating a custom monitor set "from scratch".



When a template is applied to a system, changes that are made to the template will be reflected on any system where that template is linked. With the machine selected, click on the "Unlink template" option in the Commands window. This will leave a copy of all monitors from the template applied to the system, but will disassociate these monitors from the template. Once this is done, any customization of the monitors can be made without risk of being replaced by template changes.

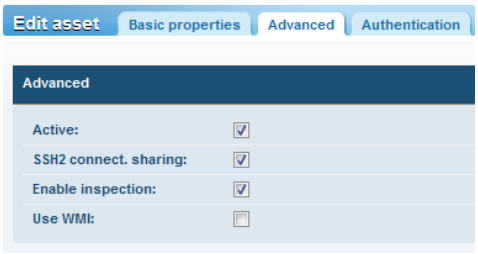
Special Configuration Requirements

Certain conditions require special configuration settings within KNM.

Disabling WMI on the Gateway Agent

The Gateway Service is supported only on Windows platforms. The limitation of the service and the Windows platform prevents it from making a network connection to the local WMI provider. To allow the Gateway system to monitor itself, WMI queries should be disabled by editing the machine settings.

- Navigate to the network folder, then click on the host where the Gateway service is running;
- Click "Edit" from the Commands window;
- Select the "Advanced" tab;
- Un-check the "Use WMI" checkbox;
- Click "Save".



Running the WMI Verify Procedure

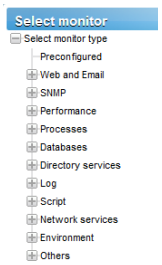
Certain Windows platforms have a security configuration applied that can prevent WMI queries from being made by the Gateway server. This will usually be presented as all Monitor and Alert type monitors reporting an alarm condition. When you click on an alarmed monitor, it will report an "access denied" or some type of "resource unavailable" message. When this condition is identified, the first remediation process should be to run the **Verify WMI for KNM Monitoring** procedure, located in the Kaseya Diagnostic Tool folder. This will update the security on the WMI service and update the local firewall to permit WMI queries from the local network.

Custom Monitors

While Core Automation provides many of the most common monitor sets, there are situations when a custom monitor may be required.

Creating a Monitor

Start by selecting the device where the monitor will be applied, then click the "Add new monitor" option from the Commands menu. The "Select monitor" menu appears. Expand the desired monitor type, select the monitor from the list, and click the "Add Monitor" button. Depending on the monitor type that was selected, various configuration parameters may need to be entered. In all cases, you will need to specify:



- Name – The name of the monitor. The name format should be consistent!
- Test interval – Usually 60 seconds for monitor-only & Operational Availability monitors, and 120 seconds for performance alerts.

- Alarm Generation – usually 15, which will be 30 minutes for monitors on a 2-minute cycle or 15 minutes for those on a 1-minute cycle. This setting will be dependent on the criticality of the monitor and the potential for spuriously crossing the threshold that you set.

If this is a monitor-only set, you might want to set the Chart Resolution to more than 24-hours. This will provide a longer review period but will result in lower fidelity of the chart.

Defining the Message Header

Custom Performance alerts will nearly always use the default message header, which is pre-configured to report performance issues. Operational Availability and other, non-performance monitors will usually require a custom alarm message header (subject) and message. The standard alarm header format is `KNM|Alarm|OA`***`type`***`-%[monitor.name]||| OA`***`type`***`.CTR.S.P3.Alm`

The parts of the subject that are highlighted are most often changed, which represent the "Data1" value that holds the alert type and monitor name, and the Type and Priority value in the Monitor Set ID. There are two optional fields – Data2 and Data3 – which are available for custom use and (obviously) follow the Data1 field. The last part of the Monitor Set ID may also be changed from "Alm" to "Act" if the alert will be associated with a remediation procedure.

Creating a Template

As discussed earlier, there are two types of templates – one for a collection of monitors and one for a collection of templates. The easiest way to begin is to create a collection of monitors for a specific application or service. A machine with no other monitors applied will be your best starting point, although any system – even with existing / unwanted monitors – will work.

Start by selecting a machine where the monitors can be created. Create the monitors you want by following the instructions reviewed previously. When the machine has the monitors you want to define as a template, click the "Save as template" link in the Commands menu.

Next, navigate to Settings / Device Templates. Locate the new template that was created (should have the machine's name as the template name). Click the Edit icon and update the template name and description. Placing an underscore in front of the name will keep all custom templates at the top of the list. Click on the template to edit the monitors – remove any that will not be part of this template.

Creating a machine template is very similar – select a machine and assign monitor templates to it. Convert the machine to a template, and when renaming the template, use two underscores to both keep the template at the top of the list and to distinguish machine templates from monitor templates.

Configuring a Dashboard

Dashboards are a handy way to view the status of critical infrastructure devices, both internal and at key customer sites.

A basic dashboard can be created during the initial setup of KNM and includes the following widgets:

- KNM System Status – displays the key information about the VSA server, including uptime, load, version, and number of monitored devices.
- Alarm Summary – shows the number of current alarms and disconnected gateways.
- Gateway Status – this is manually configured to display the status of key gateway servers. This should include the internal network of the MSP and the gateways of the top 10 customers.
- Monitor Status – This is also a manually configured list and represents the most important monitors on the most critical systems. A logical limit of about 25 monitors is recommended.

Multiple dashboards can be created and creating a dashboard for each critical customer is a good idea to get a quick understanding of the overall health of the environment. Creating dashboards is the responsibility of the MSP.

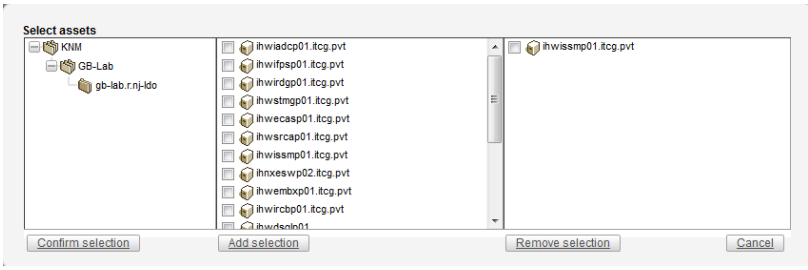
Defining Maintenance Schedules

Maintenance schedules define the days and time ranges when monitors will not generate alerts. Schedules can be defined by device (most common) and by individual monitors. The interface is a bit cumbersome as each device or monitor must be selected individually (no “select all” option).

Device Schedules

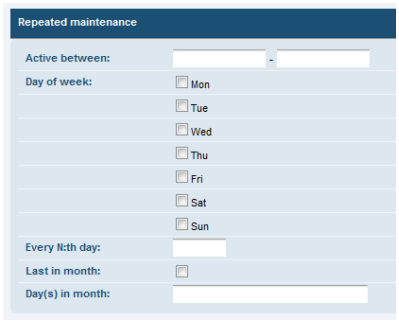
When one or more devices at a customer location will be down for an extended period, a Device Maintenance schedule should be created. Navigate to Schedules / Device maintenance and select “New Schedule”. This can be a Single Maintenance with a start time and duration or a Repeating Maintenance, which adds the ability to specify a range of dates when the maintenance will be active.

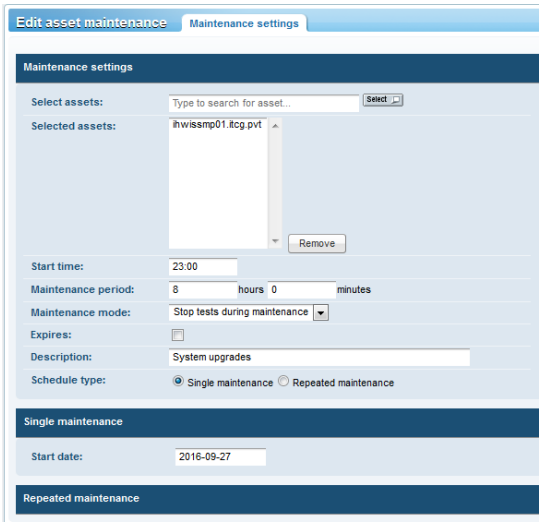
When selecting assets, click the Select button – the selection window appears:



Select the network from the first field, then select the host(s) from the second field and click “Add selection” – the device(s) selected will appear in the third field. Click “Confirm selection” to add these devices to the maintenance schedule. Repeat as needed to select devices from other networks.

The maintenance configuration screen for a single event is shown at the far right, with the option for repeated maintenance shown to the right.





Monitor Schedules

Monitor schedules prevent a specific monitor from generating alarms. This could be useful when making changes to the monitor set, or to prevent a certain set of monitors from generating alarms within a specific time range. Note that the ITP configuration (next chapter) has a method for excluding alerts by monitor type that is probably more flexible than creating a monitor schedule for most situations.

The process for defining a monitor schedule is the same as a device schedule, except that individual monitors are being selected and the exclusion applies to all devices.

VSA Configuration Overview

Understanding the logic behind monitors is essential to creating custom monitors and remediation procedures. Developing and using standards is the cornerstone of reliable automation and smooth operation of your MSP practice.

The entire RMM Suite is based on well-defined standards for organization, monitor naming and leveraging the ability for software to parse this information. *The more you can accomplish with your "silicon-based engineers", the more time your carbon-based engineers will have available for more profitable ventures!*

Customer Setup

Consistency in the agent group structure is essential for automated processing.

Customer ID And Machine Group Naming

The following guidelines are strongly recommended for all MSP environments:

- Customer ID
  - Keep it short – 15 characters or less, with no special characters.
  - Keep it meaningful – the company that the ID represents should be easily identifiable.
  - Keep it consistent – use the same ID for all applications that use customer data.
- Customer Name - this MUST be the same, including punctuation, as the name used in your PSA if API integration modules are used with ITP. As the name is generally not used in automation in VSA, it is a reasonably safe to update the name in VSA to match your PSA.
- Root Group – the root group is essential for policies to be applied to the entire organization. The root group must not directly contain any agents.
  - Use "m" for the typical root group. This keeps the name short, preventing group column overflow on displays and allows policies to be applied to the customer organization. It also identifies the client as a "managed" customer. When onboarding a new customer, you can use "audit" for the root group name to temporarily block automation while performing onboarding tasks. Rename this to "m" to enable automation.
  - Use "unm" for unmanaged (break-fix) clients. The Auto-Pilot policies will not automatically be applied to this root group name (policies can still be applied manually, if desired). This prevents monitoring, updating, and patching from being performed on these customer agents.
  - Use a short ID when a single managed customer has multiple divisions with unique settings for each division. Do not use this method to identify different customer locations!
- Machine Class – using machine class groups can help prevent policies from being mis-applied should a filter (view) be incorrectly defined. It also permits treating workstation platforms as servers (and vice-versa), which can't be done with views.
  - **servers** – any machine classified as a server is placed here. This can include workstation class systems used in server roles (such as a FAX server, phone or voicemail system).
  - **wkstns** – any machine classified as a workstation, including systems with a server O/S used as a workstation. This group name may have a numbered designation 1-3 to define alternate workstation patching configurations supported by the RMM Suite.
  - **tclients** – Thin-client workstations are placed here. Even though they can be considered as workstations, they generally require special handling for updates and have alternate administration and management methods.
  - **special** – Any machine that requires special handling is placed here. When a machine is moved into this group, all Auto-Pilot monitors are immediately removed. If monitoring, maintenance, or patching is desired, the policies must be applied directly to the machine.

(This is why Auto-Pilot policies are duplicated as manually configured policies.) When a machine is removed from the special group, the Auto-Pilot are allowed to once again take effect.

- Location – Specifies the location of the agents. The use of this field is strongly recommended, even when a customer has only one location, so that parsing of the machine.group data is done in a consistent manner. It also makes updating easier when an existing client adds a new location. Use of the UN LOCODE standard is strongly recommended as globally-unique identifiers exist for nearly every town and city in the world. Core Automation has been used successfully with a slightly modified version of the LOCODE standard.

Site ID Standards

Core Automation is delivered with a standard copy of the UN LOCODE database for North America in an Excel spreadsheet format. Use of this standard has been discussed in the introductory pages of the Operations section. The MSP is encouraged to utilize the RMM 8 model of LOCODE data from the spreadsheet provided by MSP Builder during onboarding. The RMM 6 format is no longer supported.

Special Situations

When a client has multiple locations within the same city code, a short suffix can be added. Delimit the location name from the suffix with an underscore (_) character if creating this manually, or define the Qualifier value in the CustomerLocations spreadsheet.

When servers are hosted at a third-party data center, the specific location may not be known, or may not be as important as knowing that the servers are "in the cloud". In these cases, an alternate Site ID method is used. The general format is similar to the UN LOCODE, but can indicate a regional location and provider ID instead of a specific city. The MSP can identify their own identity values so long as they are documented and used consistently. The format used by the MSP Builder team is "cc-rr_pp".

- "cc" represents a country – the standard UN LOCODE country identifier is used.
- "rr" represents a region (or state).
- "-lll" represents the 3-character LOCODE city/town code of the customer's primary location.
- "_" is the character used to delimit the qualifier.
- "pp" represents the Provider ID. Some common providers in the US are:
    - AW Amazon Webhost
    - AZ Azure (Microsoft)
    - GG Google

Creating a New Customer

There is a very specific process for creating a new customer and taking the time to follow it will ensure that all automation works as expected. Many MSPs use their PSA or accounting system to track prospects, so customer information should be set up correctly in these applications *first* to ensure consistency between all applications.

Most PSA platforms either synchronize or share data with Kaseya. This allows tickets submitted by Kaseya to be properly associated with a customer (and – usually – a registered device) in the PSA. When a new customer is defined in the PSA, be sure to complete any additional configuration steps needed to permit this association. (For example, in ConnectWise, the "Management ID" must be defined.) This underscores the importance for consistent Customer IDs in all applications. The Customer ID should be defined in the application with the most restrictive requirements first, which ensures that the ID can be the same across all platforms.

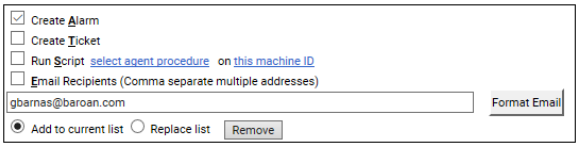
Kaseya Customer Setup Checklist

These are the common steps needed to define a new VSA client.

1. Create a new organization
    a. Use the Customer ID and name defined in the PSA.
    b. Select the appropriate Org Type (Standard, Extended, or Full) depending on their contracted coverage hours. (Some MSPs will have multiple Extended coverage options – be sure to select the correct one.)
    c. Set the default MachGroup and Department correctly. For managed customers, this is usually set to "m". If the customer is unmanaged (break/fix), use "unm" instead.
2. Create the agent machine groups:
    a. Select the newly created customer, then select the Machine Groups tab.
    b. Create groups for Servers, Special, and Wkstns. Optionally, create a TClients group if the customer employs thin-client systems.
    c. Create sub-groups under the Servers and Wkstns groups for each site where these types of systems exist. Do the same if TClients are used.

The above steps can be automated using our VSA Org Management tool.

3. Set the Naming Policy (OPTIONAL)– having the agent name match the computer name is strongly recommended.
    a. Navigate to System / System Preferences / Naming Policy
    b. Select each group associated with the new customer, enable the "Force machine ID…" checkbox, then click Update. Repeat for each customer group.
4. Add the client to the appropriate scope(s). A global scope allows the MSP technical staff to access all customer agents without being members of the Master scope, which grants additional access rights that may not be appropriate for all MSP technical staff.
   If the customer has technical staff that will utilize the VSA, a customer scope should be created and the customer added to that scope using the same process.
    a. Navigate to System / User Security / Scopes
    b. Select the appropriate GLOBAL scope (default is "Personal"), then select the Organizations tab.
    c. Select the new customer organization and click "Assign"
    (all customers should be members of the global scope!)
5. Set the managed variables for the RMM_Admin credentials to the new customer groups.
    a. Navigate to Agent Procedures / Schedule/Create, click Managed Variables.
    b. Select the RAUserID (Pub) variable, enter the appropriate User ID.
    c. Select all of the customer's machine groups, then click Apply.
    d. Confirm that the customer machine groups are still selected, then click Apply.
    e. Repeat steps b-d for all other managed variables as appropriate.
6. Configure the New Agent Installed monitor action:
    a. Navigate to Monitor / Agent Monitoring / Alerts
    b. In "Select Alert Function", choose "New Agent Installed"
    c. Click the Select All link
        i. select "Create Alarm"



    d. select Replace List option and click Apply

The following steps are optional depending on components being deployed and used by the MSP.

7. Network Monitor - Create a subgroup for the client.
    a. Create a user account for alerting - see the Alerting User Accounts page for information on creating the alerting account.
    b. Locate the site-network ID folder and move it to the new client subfolder.
    c. Select all devices not to be monitored in KNM and set the status to "Deactivate". The status should display as "gray".
    d. Apply appropriate monitoring templates to each server.

Monitor Sets

All monitors used in the RMM Suite employ a standard format that makes the information easily parsed and processed by automation software such as ITP, Service Desk, and many PSA platforms. Each monitor consists of six fields, delimited with a "pipe" symbol (|). It is important to understand the basic format of a monitor alert so that custom monitors can be created and properly parsed by the automation procedures. A similar method is used when sending alerts to the PSA, allowing advanced parsing and ticket routing.

Alert Message Format

The Alert Message is set by defining the *email subject line* in the Monitor/Agent Monitoring section. The general format of an alert created by a monitor is:

```
Source | Event | Data 1 | Data 2 | Data 3 | MonitorSetID
```

Spaces have been added to the above example for clarity – normally there are no spaces around the delimiters.

- **Source** – the source ID of the alert. Current sources are listed below, but others can be defined as required.
    - **STS** – Agent Status, such as offline, check-in, or disabled.
    - **APP** – Application Change (install, remove).
    - **HWC** – a hardware change, adding or removing a board, disk, or memory module.
    - **DSP** – Drive Space Alert – not used by the RMM package.
    - **PRC** – Procedure failed* – not used by the RMM package.
    - **PRT** – Protection Failure – an AV protection alert.
    - **NAI** – New Agent Installed / first time check-in.
    - **PAT** – Patch Event.
    - **SYS** – Kaseya System events.
    - **EVT** – Event Log Event.
    - **MON** – a Monitor Set alert.
    - **EXM** – An external monitor alert.
    - **KNM** – a Kaseya Network Monitor alert.
    - **AVM** – AntiVirus or AntiMalware alerts.
    - **PCA** – Procedure Custom Alert via the sendAlert function.

    *The "procedure failed" alert has limited value as many Kaseya-internal procedures depend upon "failing" to process decision logic. MSP Builder does not deploy monitors for Procedure Failed for this reason.

- **Event** – sets the specific event that occurred for the defined Event Source. This data is unique within each Event Source. For example, the events "Account Disabled" and "Account Locked" distinguish between two "SYS" Event Source IDs. For Event Log alerts, this will identify the event source (System, Security, or Application).
- **Data1-Data3** – provide additional information for parsing the alarm. These fields are primarily used for parsing by auto-remediation applications and ticketing systems such as BMS, AutoTask, and ConnectWise. For example, the Event Log alerts use these fields to report the Event ID, Event Source, and similar data.
- **Monitor Set ID** – is a unique identifier consisting of five fields delimited with periods. This is defined in detail in the following pages.

The standard alerts are documented in detail in the Monitor Set List document.

Monitor Set ID

Monitor Sets are named to clearly identify their purpose and assist in deployment and automatic parsing. All monitor set IDs must use the following format, and all fields are required:

`<Name>.<Class_Code>.<Group_Code>.<Priority_Code>.<ActionType_Code>`

The Monitor Set ID is the primary method of parsing the ticket in ITP, and using a unique and well-defined ID is essential to proper processing in ITP. The following page illustrates the Monitor Set ID in full detail.

Name Part	Allowed Values
<Name>	A unique name applied to the monitor set. This may be names such as "Core" to identify standard objects or a specific application or customer name. Should be 10 characters or less.
<Class_Code>	One of the following codes: SVC="Service" EVT="Event Log" CTR="Counter OAM="Operational Availability Monitor" This field is used by the ITP module and often mirrors the Source field.
<Group_Code>	One of the following codes: S="Server" W="Workstation" H="Hypervisor" R="RDS_Host" C="Custom" X="Multiple" (ie: servers and/or workstations) N="Network" This field visually aids the correct assignment of the monitor set to an appropriate target.
<Priority_Code>	One of the following codes: P1="Critial" P2="Essential" P3="Normal" P4="Low" P5="Info". This will be used to set the ticket priority and auto-remediation goal time during ticket intake processing. It is also used for advanced notification processing.
<ActionType_Code>	One of the following codes. These can generate PSA tickets: "Act" - the event can be remediated through procedures. "Alm" is an event that only generates an alarm. The following do not generate PSA tickets: "Req" - an event that relates to a request for service. “Log” – used by EMM tools to report a successful self-remediation. "Chk" is used to identify "missing events" during testing and diagnostics. "Inf" is used for non-alarming/non-remediation events, such as onboarding data collection. These are ignored by ITP. “Act” and “Req” types are associated by ITP with remediation procedures.

Note that field names are case and position sensitive!
The MSP Builder ITP service is not case-sensitive, but any tests performed within the Kaseya procedures are. It is much easier to treat all monitor set fields as case-sensitive and use a consistent form.

Standard Monitor Set IDs

The RMM Package includes a large assortment of monitors that can be used as-is, adapted and extended, or used as the basis for a set of custom-defined monitors. The most important consideration is to ensure that the Alert Message Format and Monitor Set ID guidelines be followed.

Global Alerts

These MonSetIDs are defined in the Agent Monitoring / Alerts section. Not all of the MonSetIDs are used by monitor sets but are defined should the MSP desire to use them.

MonSetID	Description
STS.X.P4.Alm	Agent Offline alert
AgentStatus.STS.X.P4.Alm	Agent Remote Control Disabled, Came online, or multiple agents offline
AppChange.APP.X.P4.Alm	Application change detected
FileChange.DAT.X.P5.Alm	Alerts associated with Get File events
HardwareChange.HWC.X.P4.Alm	Alerts associated with Hardware Changes
DriveSpace.DSP.X.P3.Alm	Low disk space monitors (agent monitor, not Smart Monitor)
APFail.PRC.X.P4.Alm	Alert when an agent procedure fails
ProtVio.X.P4.Alm	Protection Violation alerts
CheckIn.NAI.X.P4.Req	New Agent first-time checkin – runs procedure
Patching.PAT.X.P4.Alm	Alerts from the Patching subsystem
KSysMon.SYS.X.P3.Alm	Alerts associated with the Kaseya server subsystem

Service Monitors

The following MonSetIDs are used by Service Monitors.

MonSetID	Description
ADDC.SVC.S.P1.Act	Active Directory Domain Controller service monitors
Altigen.SVC.S.P3.Act	Altigen Server service monitors
BBS.SVC.S.P2.Act	BlackBerry Server service monitors
Core.SVC.S.P2.Act	Core Server service monitors
Core.SVC.W.P3.Act	Core Workstation service monitors
Core-ND.SVC.W.P3.Act	Core non-domain Workstation service monitors
DHCPSvr.SVC.S.P1.Act	DHCP Server service monitors
DNS.SVC.S.P1.Act	DNS Server service monitors
Exchange.SVC.S.P1.Act	Exchange Server service monitors
FileSvr.SVC.S.P2.Act	File Server service monitors
GFI.SVC.S.P1.Act	GFI Archiver Server service monitors
HMail.SVC.S.P1.Act	HMail Server service monitors
IIS.SVC.S.P1.Act	IIS Server service monitors
RDSCB.SVC.S.P1.Act	RDS Connection Broker service monitors
RDSGW.SVC.S.P2.Act	RDS Gateway service monitors
RDSLS.SVC.S.P1.Act	RDS Licensing Server service monitors
ShadowProt.SVC.S.P3.Act	ShadowProtect Server service monitors
SharePoint.SVC.S.P1.Act	SharePoint Server service monitors
SQL.SVC.S.P1.Act	SQL Server service monitors
vcenter.SVC.S.P1.Act	vCenter Server service monitors
Veeam.SVC.S.P3.Act	Veeam Server service monitors

"Matching" is enabled on the Service Monitors to monitor only the services found on an agent, allowing a single monitor set to contain all necessary monitors for all versions and configurations of a product.

Any monitor set defined by the MSP should follow a similar naming format and should be documented.

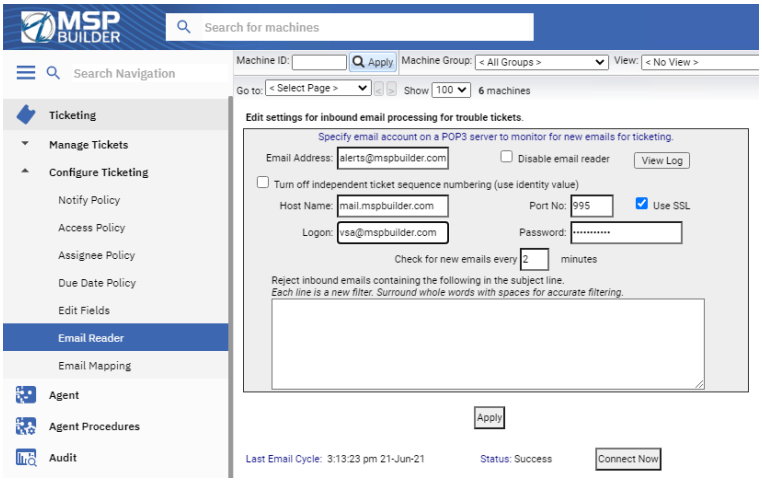
Event Log Monitors

All Event Log monitors utilize a format that identifies them as MSP Builder monitors, the Event Log that is associated with the monitor, and an identifier. See the Appendix: Monitor Details for more information.

MonSetID	Description
MB-ADDC-x.EVT.S.P2.Alm	Active Directory P2 Alert Monitors
MB-ADDC-x.EVT.S.P3.Alm	Active Directory P3 Alert Monitors
MB-ADLK-C.EVT.S.P3.Alm	AD Lockout Monitor
MB-AINS-A.EVT.S.P2.Alm	Kaseya Alert Caller (Voice Notification Service) monitor
MB-ASM-A.EVT.S.P3.Alm	Auto-Start Monitor
MB-CORE-S.EVT.S.P3.Alm	Core SYSTEM event log alerts – servers
MB-CORE-A.EVT.S.P#.Alm	Core APPLICATION event log alerts, P2 and P3 – servers
MB-CORE-A.EVT.W.P3.Alm	Core APPLICATION event log alerts, P3 –workstations
MB-CRSH-S.EVT.S.P3.Alm	Crash Detection SYSTEM event log alerts – servers
MB-DHCP-S.EVT.S.P#.Alm	DHCP SYSTEM event log alerts, P2 and P3 - servers
MB-DNS-S.EVT.S.P#.Alm	DNS SYSTEM event log alerts, P2 and P3 - servers
MB-EXC.EVT.S.P3.Alm	Exchange Server event log alerts
MB-ICC-A.EVT.S.P3.Alm	Internet Connection Failover Check alerts
MB-IIS.EVT.S.P3.Alm	IIS Web Server event log alerts
MB-KNMG-A.EVT.S.P3.Act	KNM Gateway APPLICATION event log alerts
MB-LDS-S.EVT.S.P#.Alm	Local Disk Subsystem SYSTEM event log alerts, P2 and P3
MB-MNT-A.EVT.?.P3.Alm	EMM Maintenance APPLICATION event log alerts, server and workstation
MB-NTS.EVT.S.P3.Alm	Network Time Service event log alerts
MB-PSP.EVT.S.P#.Alm	Print Spooler event log alerts, P2 and P3 - servers
MB-RDCM.EVT.S.P3.Alm	Remote Desktop Connection Manager event log alerts
MB-RDGW.EVT.S.P3.Alm	Remote Desktop Gateway event log alerts
MB-RDLC.EVT.S.P3.Alm	Remote Desktop Licensing event log alerts
MB-SM-A.EVT.?.P3.Alm/Log	Smart Monitor event log alerts and log events
MB-SM-TIME-A.EVT.S.P3.Act	Network Time Smart Monitor Remediation event log alert
MB-SQL.EVT.S.P3.Alm	SQL Server event log alerts

External / Agentless Monitoring

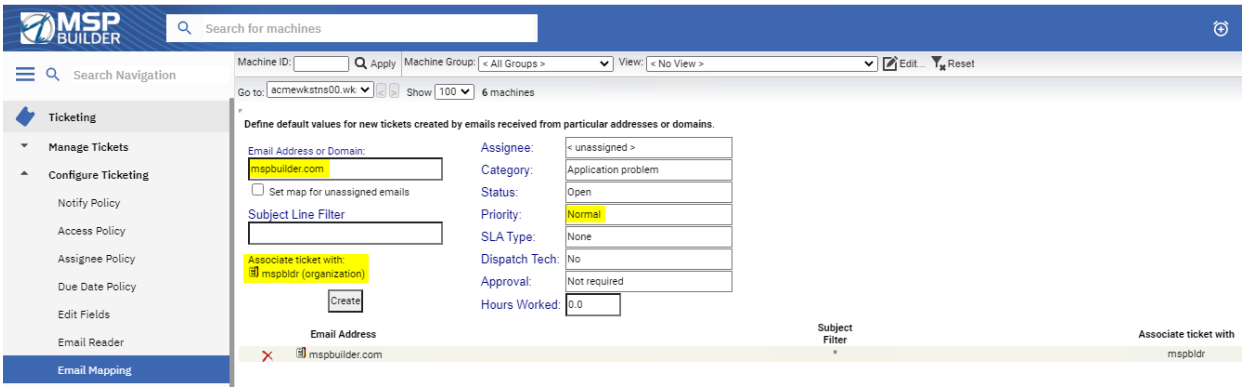
External or Agentless monitoring can be used to accept notices from devices that send email alerts. These are often network devices, such as routers, firewalls, UPSs, and printers. The RMM Suite utilizes Kaseya's Ticketing Module to receive and process these email-based alarms. For these to function, you must create a POP3 email account for the exclusive use of VSA. Navigate to Ticketing - Configure Ticketing - Email Reader and configure the settings as shown below (using your email address, host, and domain - of course!)



Once this is done, you can send an email to that address and verify that it is received as a ticket. You can delete that test ticket once you verify that it has arrived.

Mapping Email Domains to Clients

Since emails do not directly identify the device, and often don't even identify the customer, VSA provides the ability to map emails and email domains to assets and organizations. When a domain is mapped to an organization, the RMM Suite can deliver the alert to the correct organization in the PSA.



For most situations, mapping the email domain to a VSA organization will be sufficient.

Review the ITP Configuration Guide for details on rewriting the subject to allow further processing and classification to be performed.

MSP Builder Procedure Library

The RMM Suite comes with an extensive set of procedures. Even though we have tried to provide as diverse a selection of procedures as possible, MSPs will always need to create their own to accommodate their unique situations. Let's start with looking at recommendations for creating additional procedures to extend the library.

Creating Procedures

Most users should have the ability to create and edit their own procedures. These procedures should be reviewed and approved by a member of the NOC Management team prior to actual use. This review should ensure that best practices are followed, including a complete description field, code comments, and use of standard methods and folder locations. Setting and maintaining code standards takes effort, but the few minutes it takes to follow these standards will pay dividends in reliability.

Basic Guidelines for Creating Procedures

- Make sure that the Procedure Summary Description is filled in. It should describe the procedure's purpose in a general manner.
    - When editing the procedure, click on any blank space. Be sure that the right side window is expanded. The Description field will be displayed and can now be filled in.
- Use comments in the procedure to explain the purpose of each block of commands. This will make it easier to review/approve as well as maintain the procedure by others in the future.
- Use the system temp folder for copying installation files, not the KWorking folder. There is a sample install file that illustrates this process and can be used as a starting point for new procedures.
    - If C:\Temp is not present, create it (Windows only).
    - Create a subfolder in the temp folder to hold the install files.
    - Copy the install files to the subfolder created above.
    - Run the installer from the temp/subfolder, writing any logs to this folder.
    - If desired, upload the logs to Kaseya.
    - Remove the temp\subfolder (recursively) to reclaim the used space.
      Note that the temp folder will be "/tmp" on Mac/Linux platforms, and "C:\Temp" on Windows Platforms.
- Create the procedures in your personal folder. After requesting a review and receiving approval, test the procedures to verify that they work as intended. If the procedures are ready for general use, inform the NOC management team and the procedure(s) will be moved from your personal folder to an appropriate shared location, and the procedure documentation updated.
  Maintaining private procedures for production use is not recommended and can result in duplication of coding effort!

Sample Procedures

There are sample procedures located in the _MSP Builder/Core Automation/Application Procedures folder that illustrate many common tasks. These can be used to create new procedures based on standardized methods.

Sample_Install

This procedure illustrates a typical installation process that copies installation files to the local machine, runs the installation process, collects log files, and then removes the install folder from the temporary location. The commands can be removed if not needed or modified to suit the specific requirements. The key steps performed include:

- Prompt for input – the example asks for a license key.
- Check for C:\Temp folder and create it if not found.

- Create a sub-folder in the Temp folder.
- Write a message to the procedure log indicating that the procedure is starting.
- Copy files and / or folders to the installation temp folder. The example also shows how to copy platform-specific (32 or 64-bit) files and folders.
- Execute the installer command.
- Use GetFile to collect the installation log files.
- Remove the installation temp folder structure.

Sample_ManagedVariables

This procedure demonstrates how to use Managed Variables. Managed Variables contain global or customer-specific values in globally named variables. These variables can be used directly by procedures, but a procedure may fail if the managed variable is blank or not defined. The sample procedure uses a second procedure to return the Managed Variable value if it is defined. If the second procedure fails, the default value (as defined on line 4) is used.

The sample specifically checks for the default value being present and aborts the process if customer data isn't available. This can be adjusted to define a workable default value that will be replaced if the Managed Variable is defined. The resulting variable is simply used to perform the task or apply the value. The key to this sample is to avoid errors when the variable is undefined.

The sample procedure is shown here.

```
1  // Get the specified managed variable value or return xFALSEx if not defined
2  // Define the default value to use if the managed variable is undefined
3  // Leave defined to "xFALSEx" if a specific default is not required
4  getVariable("Constant Value", "xFALSEx", "global:MVarVal", "All Windows Operating Systems", "Halt on Fail")
5  // Call the proper procedure to return the data in a variable called "global:MVarVal"
6  executeProcedure("ALL-GetManagedVar-MGUIArgs", " ", "Immediate", "All Operating Systems", "Continue on Fail")
7  // .
8  // Option 1 - alter the command based on specific variable data being available
9  // Perform either a generic or specific command
10 If checkVar("#global:MVarVal#") Contains "FALSE"
11     // The default value is defined, so run a generic command
12     sendMessage("Performing the default action", "Display now", "All Operating Systems", "Halt on Fail")
13 Else
14     // The specific variable is defined, so run a specific command
15     sendMessage("Performing specific action #global:MVarVal#", "Flash icon", "All Operating Systems", "Halt on Fail")
16 // .
17 // Option 2 - utilize the data directly with the default value defined in line 6 or the actual managed var data
18 sendMessage("Variable contains: #global:MVarVal#", "Flash icon", "All Operating Systems", "Halt on Fail")
```

NOTE: *The above procedure is used to overcome a VSA software defect that causes the procedure to crash if the Managed Variable is not defined for the machine group where the agent procedure is run against. This method is no longer required as of version 9.5.0.19 but is still used for backward compatibility.*

NOTE: *To ensure that custom procedures are preserved during RMM Suite Upgrades,* ***ALWAYS*** *place your custom procedures into a folder structure separate from the _MSP Builder folder.* ***NEVER*** *modify or reference MSP Builder procedures – always duplicate them and place the copy into your custom folder! When the MSP Builder RMM Suite is upgraded, all components may potentially be removed and replaced with the new versions.*

Deploying and Executing a Utility

When a process must run multiple times, or on a scheduled basis, the preferred method is to create a script that is maintained on the Kaseya shared files folder. This script can be copied to the KWorking directory and then executed. Copying the file prior to each execution may seem repetitive but it ensures that the most current version is always used.

The typical procedure would be something like:

- Use writeFile to copy a specific script file to the KWorking* folder. The script should use command-line arguments to prevent embedding confidential information such as credentials in the script file.
- Use a variable* to represent the name of the actual KWorking directory, which will allow the procedure to run on any Kaseya platform.
    - Place the script into an appropriate folder on the KaseyaVSAShared folder (K: drive)
        - _Customer Files folder for customer-specific files. Each customer should have their own subfolder here.
        - BAT_Scripts for any batch files (.BAT or .CMD)
        - PS_Scripts for any PowerShell scripts
        - VBS Scripts for Visual Basic scripts
        - Applications for installing/removing specific applications. Each app has its own subfolder.
- Run the procedure with the executeShellCommand function. Be sure to pass any needed variables.
    - If you need to use pipes or redirection, the command should begin with "%COMSPEC% /c" to invoke the command interpreter.
    - If you need to collect the output of the command, use the executeShellCommandToVariable instead.
    - Remember that redirection must be escaped – ">" is ">>" and ">>" is ">>>>"!

*Always reference the KWorking folder path as "#vAgentConfiguration.AgentTempDir#" or define a short variable name with getVariable("Agent Working Directory Path", "KWork") – never by hard-coding the folder name.

General Concepts for Shared Procedures

Shared procedures are those that are available to the MSP's technical staff and – potentially – customer technicians. These procedures should be run from one of two folder locations:

- _MSP Builder/Core Automation - this is the folder where most general-purpose procedures are located and can be used freely on any client system. These procedures are provided and maintained by MSP Builder.
- Customer Procedures - organized by client. This folder contains procedures that perform customer-specific tasks. This may include management of customer-specific applications, or general tasks that use customer-specific processes. When these are employed, it is preferred to use Managed Variables for the credentials or custom parameters so a single procedure can be used by multiple clients.
- Procedures located in the EMM Procedures folder should only be executed by or under the direction of the MSP Builder Support Team. These are complex procedure sets and most are not designed for general execution. These should never be shared with customer technicians! The only general purpose procedures in the EMM Procedures section are located in the Maintenance Utilities folder and are used to customize the EMM components.

Recommendation for Creating Custom Procedures

While we encourage you do use the MSP Builder tools to create customized solutions, we strongly recommend that you follow these guidelines:

- Never modify an MSP Builder procedure! Always edit and immediately Save As to make a new copy for your use.
- Store your customized tools in a separate root folder, such as **_MSP Customized**.

When the RMM Suite is upgraded, all old components are deleted, and new versions imported in their place. If you customize the MSP Builder tools or place your customized versions in the MSP Builder folder, the procedures will be lost during a platform update.

Even if you keep your customized tools in a separate folder, be sure to review them after any RMM Suite upgrades. If you reference other MSP Builder procedures, you may need to edit and re-reference the proper procedure if the new tools result in different procedure GUIDs. Kaseya procedures reference other procedures by a GUID and not the procedure name to be able to distinguish between multiple procedures that might have the same name. We always recommend that you use copies of our tools, and never reference our procedures or views in your custom processes.

Core Automation Suite Standard Procedure Library

The procedures included with the RMM Suite are documented here. As noted earlier, every procedure name starts with the platform that it supports – ALL, WIN, MAC, or LNX. Any procedure that utilizes a Managed Variable is also identified with a "(MV)" suffix.

Agent Daily Tasks

Two procedures that perform daily tasks on Server or Workstation platforms, then run the third procedure to do the actual work. These run daily audits, deploy core configuration files, and initiate the Maintenance and Smart Monitor components.

Run daily by system procedures, early AM for servers and between 10 AM and 4 PM on workstations.

Agent Init

A set of 5 procedures used for automated initialization of agents when they first check in. Step 1 identifies a Thin or Thick client, invoking the ThinClient Prep procedure for thin clients. Step 2 performs all of the MSP Builder standard initialization tasks. Step 3 invokes the RMUOBA.BMS script to run agent procedures to perform MSP and client-specific customization tasks. See the Administration section later in this guide for configuration details. The ThinClient prep procedure schedules the procedures needed to disable and later re-enable FBWF.

Agent Offboarding

This procedure used to remove all MSP and MSP-Builder components from an agent prior to it being decommissioned, including local accounts, the agent software, and support tools.

Application Procedures

A diverse collection of procedures that install, remove, or update application software packages. There are several folders that help organize the procedures by category.

Antivirus

These procedures support the standard Kaseya AV and AM components.

App Installers

Several procedures that install applications either from files stored on the VSA platform or downloaded from the vendor web site. The MSP is responsible for maintaining the URLs and downloaded content. We recommend using a generic name for files that are downloaded, whether stored on the VSA or downloaded directly from vendor URLs as this will minimize the need to customize the rest of the installer procedures. *MSP Builder provides a Zip package of the installer scripts, plus all current EXE/MSI component files. The MSP must download the zip files, extract them, then upload them to their VSA's Manage Files structure. These files can be downloaded from the www.mspbuilder.com website after logging in with a customer-enabled account.*

Backup Exec

A collection of procedures used to install, remove, and maintain the Backup Exec application.

Ninite App Management

These procedures are used for direct installation, removal, and updating of applications as well as weekly or monthly scheduled application updating scheduled by Auto-Pilot policies. This requires an MSP-provided and licensed copy of Ninite Pro to be placed in the MSPB\Support\NinitePro folder, renamed to RMMNinite.EXE.

Each procedure has an "install", "remove", or "update" prefix to clearly identify its purpose. These procedures define a variable with the name of the configuration file. A second variable enables the use of

a local cache share if it is defined as a managed variable. It then invokes the WIN-Ninite Install Utility (MV) procedure to do the actual work.

The Install Utility verifies that the variable defined in the previous procedure contains a valid configuration file name. If it does, it prepares the agent by removing old log and configuration files, deploying the current configuration file, and ensuring that the RMMNinite.EXE file is present before finally executing the RMUNNU.BMS script. This is a custom MSP Builder utility that loads the configuration file and generates the proper Ninite command line based on the configuration file settings.

Creating a Custom Ninite Procedure

Creating a custom Ninite procedure is a straightforward task:

1. Find an existing procedure type – either Install, Remove, or Update – that matches the task you want to perform. It does not matter which procedure you choose.
2. Edit the original procedure and immediately choose "Save As", changing the product name as appropriate.
3. Update the first getVariable line and change the value to reflect the product as appropriate. For example, if you chose the "Install – Chrome" as your source file and wanted to install "Itunes", simply change the value from "Install-Chrome-NiniteUpdateCfg.ini" to "Install-Itunes-NiniteUpdateCfg.ini". Save and close the procedure.
4. Navigate to the MSPB_Core\Ninite folder, edit a configuration file similar to your needs and save it with the name you defined in step 3
   (Edit "Install-Chrome-NiniteUpdateCfg.ini" and save as "Install-ITunes-NiniteUpdateCfg.ini")
5. Locate the "AppList=" line and replace the value with the name of the product you want to manipulate. Check the Ninite.com supported app list page for supported products. DO NOT enclose the product name in quotes when entering it as an AppList parameter.
6. Save the file.

That's all it takes to create a new Ninite-based application management tool.

Office 365

These procedures deploy the generic Office 365 installer program and configuration XML files that will deploy different configurations of Office 365 products. The EXE and XML files are located on the VSA in the VSASharedFiles\Applications\Office365 folder. These files are provided during installation but must then be maintained/updated by the MSP.

Outlook

A set of utility procedures to configure Outlook.

StorageCraft

Several procedures that install, update, and manage the various StorageCraft backup products.

Webroot Tools

A set of procedures to install and configure the Webroot antivirus products.

Basic Maintenance

Several procedures that perform basic maintenance tasks for MAC and Windows platforms. The Windows procedures are not provided when the EMM Suite is subscribed.

MSPB Diagnostic Tools

Procedures used to diagnose or resolve problems with VSA components. Currently, this includes a procedure that updates the WMI security settings to allow KNM to function properly, two license authorization tools, and an agent redeployment procedure that will actively replace the Kaseya agent for migrations or when outdated agents from a different VSA are found.

Policy Management

These procedures are used to either Add or Remove a "Policy Blocker" code from the Policy Management custom field. Policy Blockers are used to control when an Auto-Pilot policy should not be applied to a specific machine. There are distinct Policy Management functions, including:

- ALL Allow or block all policy blockers
- APPUD Allow or block Application Update policies
- BLMON Allow or block Baseline Monitor policies
- DAILY Allow or block Daily Task policies
- EXMON Allow or block Extended Monitor policies
- MAINT Allow or block running of Daily Maintenance
- PATCH Allow or block Patching and Software Management policies
- SMON Allow or block deployment of Smart Monitors

The blocking terms can be added for 30 days, which causes the corresponding "remove" procedure to be scheduled to run 30 days in the future. Individual monitor set blockers can be added or removed by using the "Specify" procedures. This allows blocking individual monitor sets such as "IIS" or "SQL".

Reboot/Shutdown

A set of procedures to reboot, reboot with a grace period, or shutdown any agent platform. There are also two Windows procedures that either Block or Allow the use of Shutdown from the Start Menu. These procedures can be used to ensure that agents are not shut down at night to allow after-hours support, maintenance, and patching to be performed.

Suspend Alarms

A set of procedures that suspend alarms on all agent platforms from 30 minutes to 8 hours, plus a procedure to "unsuspend" the suspended alarms. These are often called from other procedures that perform updates or reboots to prevent alerts that might result from planned work.

System Audit

These procedures comprise two distinct Audit tools developed by MSP Builder.

ALL-MB Audit -- Full System

This set of procedures is run monthly to collect information that is useful but does not change too often. Much of the information is gathered into data files and uploaded into the agent storage folder and accessible via Manage Files.

WIN-MB Data Query – Collect

This is an audit process that runs daily, collects several hundred data points from multiple locations, and updates the agent custom data fields. Many of these fields are then used to automate the deployment of monitors, scheduling of patches, and other tasks via System Policies. The procedure will use VSA APIs to update the custom fields where possible. The Update Kaseya procedure supports environments where the API is not available. This is a global selection – APIs can be enabled or disabled per VSA.

System Utilities

A large collection of utilities that perform common administrative tasks.

ALL-Force Agent Full Check-In

As the name implies, forces a full and immediate check-in process.

ALL-MB Deploy Common Files

Used to deploy all local utilities to an agent. This also invokes the MSP Builder Tool Update command, that downloads missing and outdated MSPB utilities (BMS files and supporting EXE and DLL files). This procedure is invoked daily via automation and generally does not need to be run manually.

ALL-Notify When Agent Comes Online

Run this when a system is offline and you want to know when it becomes available. Prompts for a short message (ie - a ticket # reference) to include in the message. The procedure will run when the system comes online, sending an email that identifies the host and includes the message you entered.

ALL-Set Common Data

This procedure defines variables used by several other procedures. This allows the variables to be defined once but used by many other procedures. Some parameters are written to the agent for local configuration and used by other MSP Builder agent tools. Some customization is possible in this procedure.

ALL-Update Kaseya Agent / ALL-K-Agent Upgrade

A pair of procedures that automate the agent upgrade process.

MAC-Active Directory – Join Domain

Binds a Macintosh platform to a Windows Active Directory

MAC-Computer Name – Change

Prompts for and then changes the name of a Macintosh platform.

MAC-Repair Disk – Permissions

Resets the disk permissions to defaults.

MAC-Repair Disk – Volumes

Performs a repair (fsck/diskutil) of all volumes.

WIN-Active Directory - Disjoin from Domain - REBOOT

Disjoins the computer from the current domain and joins it to the desired workgroup. The engineer is prompted for the name of the workgroup to join, which defaults to "WORKGROUP".

WIN-Active Directory – Offline Domain Join <target>

Two procedures that work together to perform a domain join without direct access to the client domain controller. The host must at some point be able to connect to the customer network to complete the process.

Only the "HOST" procedure should be invoked manually. This will locate the agent running on the client's domain controller and run the "DC" procedure. This performs the Offline Domain Join and returns the data file for use on the local host.

WIN-Add KWork to Windows Search

Adds the contents of the KWorking folder to the Windows Search service.

WIN-Agent – Set Time Zone (MV)

Sets the workstation's Time Zone value based on a Managed Variable.

WIN-Computer Name – Change [– Reboot]

Prompts for the new name, then sets the name. The other procedure does the same & reboots the system.

WIN-Config – Set DelayedDesktopSwitchTimeout

Sets the Delayed Desktop Switch Timeout to 0 for improved logon speed. The default value of 30 will display the welcome screen for the full 30 seconds after startup or logon. The purpose is to hide certain startup tasks and messages (such as the login script!). Setting this value to 0 makes all initialization visible and allows the system to be used as soon as these tasks complete.

WIN-Deploy Login Script Files

Deploys the Universal Login Script files from VSA to the Temp folder for manual configuration.

WIN-Desktop – Create Log Off Shortcut

Places a shortcut on the desktop to perform a Log Off. Usually deployed to RDS Hosts and systems where the Shutdown option has been removed from the Start Menu.

WIN-Disk – Collect Utilization

This utility deploys the Windows version of the Linux "du" utility, performs a query, and then returns the results in a log file.

WIN-Disk – Create Volume Report

Performs a disk data analysis and returns a CSV file of utilization stats.

WIN-Disk – Run ChkDsk – REBOOT

Starts by alerting the user to log off, then runs CHKDSK with arguments to run a repair at the next reboot. The system is rebooted 5 minutes after displaying the user notice.

WIN-Event Log – Export <log> and Upload

Performs an export to EVTX format of the specified Event Log and then uploads it to the VSA for further review and examination.

WIN-Firewall - <action>

Modifies the Windows firewall according to the defined action, either Enable or Disable.

WIN-IP Configuration – DHCP Enable

Configures the agent to utilize DHCP for its network interface configuration.

WIN-IP Configuration – DHCP Renew

Forces the system to perform a DHCP Lease Renewal process.

WIN-IP Configuration – IPV6 Disable

Disables IPV6 protocol on the network interface.

WIN-Last Login 1 – Save Login
WIN-Last Login 2 – Revert Login

This pair of procedures should be used when taking remote control of a workstation when the user is not present. The Save Login procedure saves the last logged in User ID to a temporary location, and the Revert Login procedure restores the saved setting. This minimizes any issues from leaving a technician's ID in the last User ID field of the login prompt.

WIN-Local Account – Administrator <action>

Enables or Disables the local "Administrator" account.

WIN-Local Account – Create ADMINISTRATOR (Prompt)

Prompts for credentials and then creates a new local user account and adds it to the Administrators group. If the account already exists, then it simply updates the password for the account.

WIN-Local Account – Create USER (Prompt)

Prompts for credentials and then creates a new local user account. If the account already exists, then it simply updates the password for the account.

WIN-Local Account – Delete (Prompt)

Prompts for the name of a user account and then deletes it if it exists as a local user.

WIN-Local Account – Delete RMM_Admin Account (MV)

Deletes the local admin account specified by the RAUser Managed Variable if it exists as a local user.

WIN-Local Account – Remove "Admin"

Removes the local account called "Admin" that is often created during system setup.

WIN-Local Account – Remove Unauthorized Admins (MV)

Checks for local administrators, removing any that are not "administrator" or specified in the RAUserID or CAUserID Managed Variables.

WIN-Local Account – Report Unauthorized Admins (MV)

Checks for local administrators, identifying any that are not "administrator" or specified in the RAUserID or CAUserID Managed Variables. Returns a file with a list of unauthorized local admin accounts.

WIN-Local Account – Set AutoLogon (Prompt)

Prompts for account credentials and then configures the system to Auto-Logon using those credentials.

WIN-Local Account – Set/Create Cust_Admin Password (MV)

Creates a new local user account using the CAUserID and CAPassword Managed Variables and adds it to the Administrators group. If the account already exists, then it simply updates the password.

WIN-Local Account – Set/Create RMM_Admin Password (MV)

Creates a new local user account using the RAUserID and RAPassword Managed Variables and adds it to the Administrators group. If the account already exists, then it simply updates the password.

WIN-Local Account – Set/Create RMM_Admin Password (MV) (Prompt)

Prompts for a password and then creates a new local user account using the name specified by the RAUserID Managed Variable and adds it to the Administrators group. If the account already exists, then it simply updates the password for the account. Used to manually override the RAPassword value.

WIN-Log Off – All RDS Users

Immediately initiates a log-off of all users on the target agent.

WIN-Log Off – RMM_Admin User (MV)

Immediately logs off the user account specified by the RAUserID Managed Variable.

WIN-Log Off – Specific User (Prompt)

Prompts for a User ID, then forces that user account to be logged off of the target system.

WIN-MB Set XEMS Status
This procedure sets the global XEMS status, disabling it if an unmanaged or audit customer is targeted.

WIN-MB Tools – Clear Manifest
This clears the local manifest timestamp, forcing a complete download of all external tools.

WIN-Patching – Set Server Schedule (Prompt)
This procedure prompts for and then sets the server Patch Schedule code.

WIN-Power Profile – Set Battery Power-Save Mode
Sets the “On Battery” power profile to “Power Save” to minimize power consumption.

WIN-Power Profile – Set Battery Presentation Mode
Sets the “On Battery” power profile to “Presentation” mode to prevent display power-off, sleep, and hibernate actions from activating. Used when presenting without a power adapter.

WIN-Print Queue – Clear
Clears the print queues, removing hung jobs from Windows Print Servers

WIN-PStools – Accept Eula
Locates all PS-Tools executables and executes them with the “AcceptEULA” parameter.

WIN-Service – Enable & Start Security Center
Sets the WSCSVC service to Automatic Start and then starts the service.

WIN-Service – Restart KNMG
Restarts the Kaseya Network Monitor Gateway service.

WIN-Service – Restart Service (Prompt)
Prompts for a service name, stops it and then restarts it after a brief delay.

WIN-Service – Set Account Credentials (Prompt)
Prompts for and sets the Service Account Credentials.

WIN-Service – Start Service (Prompt)
Prompts for a service name and starts it. No changes to the service start state are made.

WIN-Service – Stop Service (Prompt)

Prompts for a service name and starts it.

WIN-Start Menu – Set Power Action to Logoff

Changes the default power action from “shutdown” to “log off”.

WIN-System Time - Set (Prompt)

Prompts for the current time and sets the local agent time.

WIN-UAC - <state>

Enables or Disables UAC on the target agent.

WIN-User Account – Elevate User’s Rights – 15 Minutes (Prompt)

Adds the specified user to the local Administrators group and forces a logoff. After 15 minutes, it removes the account from the local Administrators group and initiates a logoff with a 10-minute grace. This allows the user local admin rights for a specific period of time.

WIN-User Profile – Cleanup - #-days – Report Only (Prompt)

Reports on all user profiles that have not been accessed in a specified number of days. The technician is prompted at run-time to specify the number of days. Any profile with “admin” in the name will be excluded from the report.

WIN-User Profile – Cleanup - #-days

Two procedures that remove any local user profile that has not been accessed within the past 60 or 90 days. Useful when run as a scheduled procedure on RDS Farms, especially with a highly transient user base.

WIN-User Profile – Delete Specific Profile (Prompt)

Prompts for a User ID and then deletes any local profile on the target system matching that account name.

WIN-W32Time - Set Configuration

This procedure sets the W32Time service configuration based on best practice standards.

If run on a PDCe, it uses NTP and configures 3 external NTP hosts, otherwise it configures the server to use NT5DS. This procedure depends on the RMM tools being previously deployed. If you are outside of the USA, please request a region-specific update!

DO NOT RUN this procedure if a client has a defined internal NTP infrastructure!

WIN-WiFi Config - Profile Data - Collect

Extracts all defined Wi-Fi profiles from a target machine, adds them to a Zip file, and uploads them using a GetFile operation. The Zip file should then be extracted to the Kaseya server's VSASharedFiles share, in a _Customer Files\<CustID>\WiFi folder. The "<CustID>" must match the Kaseya customer ID. If one of the profiles should be activated by default during deployment, then it's ".XML" file extension should be changed to ".DEF".

WIN-WiFi Config - Profile Data - Deploy

When run against a target system, it identifies the Kaseya customer ID. It then copies the contents of the VSASharedFiles share _Customer Files\<CustID>\WiFi folder to the system. All .XML files are loaded as Wi-Fi profiles. The .DEF file, if present, is loaded as a Wi-Fi profile and then activated.

WIN-Win 10 –Fast Boot - Disable

Disables the Fast Boot option on Windows 10 platforms. When enabled, Fast Boot will prevent the uptime from being reported properly.

Thin Client Support

A suite of procedures for managing the configuration, application installation, and security of a thin client system are provided, and will usually require some amount of customization prior to use. In particular, the location of the FWBF scripts (default is the user desktop “Admin”) may need to be updated.

Begin any thin-client configuration with the **Thin Client 1** procedure. This will be followed by any custom procedures or manual update tasks. Most procedures will follow the naming format "Thin Client X", where "X" is a letter. The letters currently don't have any specific meaning but can be used to group similar procedures together. The actual task performed by the procedure is defined in the procedure name after the “Thin Client *X* –” part of the name.

When all customization tasks have been completed, run the Thin Client 2 procedure. This will automatically invoke the Thin Client 3 to re-enable FBWF and reboot.

WIN-Thin Client 1 - Disable FBWF - REBOOT

Disables the FBWF to allow writing to the flash, then reboots. After reboot, the system will be ready for any updates and configuration items to be applied.

Any manual tasks and other install/update procedures should be run at this point. Once all customization is complete, the Thin Client 2 procedure should be run to complete the updates and re-secure the system.

WIN-Thin Client 2 - Update FBWF Scripts - REBOOT

Updates the FBWF Admin scripts, then Reboots

Call this AFTER any custom updates to thin clients

This procedure will call Step 3 to Enable FBWF and reboot.

WIN-Thin Client 3 - Enable FBWF - REBOOT

Runs the FBWF_Enable.BAT script (which reboots)

DO NOT RUN THIS DIRECTLY - it is scheduled from the Thin Client 2 - Update FBWF Scripts REBOOT procedure.

WIN-Thin Client A – Thin Client Install

This intermediate task invokes one or more sub-procedures to automate a thin client machine setup.

WIN-Thin Client U – WE7 Install Updates

This intermediate tasks invokes one or more sub-procedures to apply specific updates and patches.

WIN-Thin Client X – Lenovo WE7 Enable Installer

An intermediate task that enables the Windows Update and Windows Installer service on WE7 Lenovo Thin Clients.

There are several “Thin Client X -” procedures that perform general setup tasks. Many of these are called from the previously documented procedures. Having many separate setup procedures allows one, several, or all to be invoked.

Administration

In this section, we will review the administration tasks and troubleshooting methods needed to operate and maintain the RMM Suite.

Patch Management

Monthly, Second Week

Review and approve or decline patches by policy on the Thursday or Friday **following the Second Tuesday** of each month.

- Navigate to Patch Management / Patch Policy / Approval by Policy
- Select each policy and review the patches that are Pending Approval.
    - Deny any patches that match the policy criteria
    - Select all remaining patches and approve.
    - Repeat for each Patch Policy

Monthly – Spot-Check

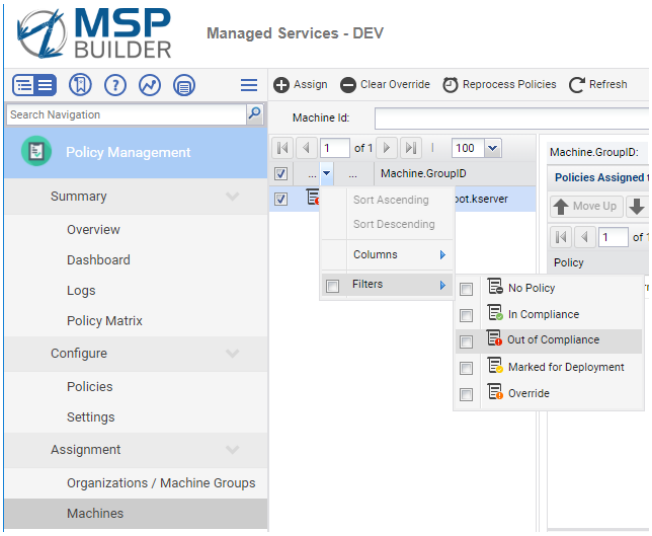
Check that patch scans and update schedules are being applied to agents as expected. Check one or two customers each month, and new customers after onboarding.

Policy Management

Ensure that policies are being applied properly, and deploy policies that override the defaults assigned by the Auto-Pilot policies. Apply “Policy Blockers” as necessary to prevent Auto-Pilot settings. Define custom settings to apply patch schedules.

At Least Weekly

Verify that System Policies are being applied and enforced.



- Navigate to Policy Management / Assignment / Machines
- Apply the Filter first for Override and then for Out of Compliance - do not apply both together as Out of Compliance will mask agents in Override state.
- Review the results and examine any non-compliant systems.
    - Overrides indicate that some manual setting or monitor has been applied. Review and apply a proper policy, then clear the overrides.
    - Non-Compliant systems usually indicate a conflict between policies, or an invalid setting in one of the policies. Review the policies, adjust, and then re-apply the policies.

As Needed – Custom Situations:

These tasks are required when a customer group or agent requires custom settings that differ from the Auto-Pilot policies.

Apply a Custom Policy to Override Auto-Pilot Settings

There are many policies provided that can override the Auto-Pilot policies. These can be used directly or to create custom policies of your own. To create a custom policy, select an existing policy that matches your requirements as closely as possible, click **Save As**, and provide a new name. DO NOT use Auto-Pilot policies as a template! Make the changes, then click **Save and Apply**. We recommend that you move the new policy to a folder outside of the _MSP Builder structure so custom policies are not impacted in future updates.

If you want to apply an override policy to a machine, you don't need to specify a view. To apply policies to machine groups, you must specify a view. There are several "XPOL-Unrestricted" views that can be used. For extra protection, these can be used to create customer-specific views – replace "unrestricted" with the customer ID when copying these views.

To apply the overriding policy:

- Navigate to Policy Management / Assignment / Organizations/Machine Groups
- Locate the Machine Group where the overriding policy should take effect. You may need to expand the customer folder to apply the policy to a type or site-specific group.
- Expand the Policies tree on the right side and locate your Override Policy.
- Drag the Override Policy and drop it on the desired machine group. Note that if you accidentally drop it on the wrong group, you'll need to remove the policy from the group and drag it again to the correct group.
  We suggest that you hover momentarily over the group folder icon before releasing the mouse key.

Apply a Policy Blocker to Prevent Auto-Pilot Settings

Sometimes, you simply need to prevent an Auto-Pilot policy from running on a specific machine or group of machines. A custom view isn't needed in these cases. Policy Blockers are the simple solution to these situations.

Each agent has a custom field called "Policy Control", which is normally empty. By running one of the Policy Control procedures, you can selectively add or remove values from this custom field. These prevent Auto-Pilot policies from being applied. For example, a customer has a development server and does not want to receive any special alerts for applications, but still wants basic monitoring. By running the Policy Control procedure to apply the "EXMON" blocker, all normal maintenance and monitoring will be performed, but Extended monitors will not be applied.

This can be even more direct by using the Policy Control – Specify procedure. In this case, "SQL" can be provided as an argument, and only SQL monitors will be excluded from that agent.

Do not edit the Policy Control custom field directly, as the values are stored in the agent's registry and sync from registry to custom field during daily audit process.

Define a Server Patch Schedule

Server patching requires some extra effort to ensure that systems are rebooted in the correct order, so applications are started properly. MSP Builder provides a large selection of Auto-Pilot policies that configure the monthly patching but depend on a manually defined setting. Each system has a Patch Schedule custom field and entering a value into this field will cause the corresponding schedule policy to be applied.

These values can be defined via the Audit tab, editing the custom field Patch Schedule manually. A procedure – "WIN-Patching – Set Server Schedule (Prompt)" – can also be used to set the same schedule code on several systems at once.

We recommend the use of our “Offline Patch Management” tool, which can be downloaded from the MSP Builder website and installed on any *locally* shared folder. This provides a simple, Excel-based interface to manage server patch schedules.

Audit Customization

The MSP Builder Daily Audit tool collects over 200 parameters each day. Some of this data is uploaded to custom fields. One in particular is the System Roles custom field, which contains tags for each of the system roles, features, and applications that were detected during an audit. A common requirement that MSPs face is to add detection for applications that are not in the standard audit table. Generally, an MSP will need to define a custom detection, then create a monitor set for the application, and then create a System Policy and View to create an Auto-Pilot applied monitor set.

NOTE: *Do not use this functionality for collecting data for reporting. The custom field data length is limited and collecting extra data in the System Roles can overload the field and prevent monitors from being deployed or maintenance tasks from being run. In versions 2.5j and higher, a separate custom field can be defined named "System Data", and the sections MSP SERVICE ROLES and MSP APP VERSION ROLES needed for reporting can be created and defined using the same data format. These detections will populate the System Data field instead of the System Roles field.*

Create a Custom Detection

- Locate the **AuditRoles.ini** file in the **VSASharedFiles\MSPB_Core\Root\Audit** folder on your VSA host. Open the file using Notepad or a similar text-only editor.
- Scroll down to the [SERVICE ROLES] section. This section defines a mapping of Service Name to Identity Code. The left side is the name of a service, or the partial name of a service if multiple services exist.
    - Create an entry for the service that you need to detect.
    - Verify that it does not already exist or conflict with an existing service name. The service name should be as specific as possible without including instance-specific values.
    - Assign a 3 to 4-character code to the service. Search the file for your code to make sure it isn't already used.
- The [IGNORE SERVICES] section can be used to ignore certain services. For example, you might want to detect the "Framis" service on servers but ignore the "Framis-Reader" service that is found on client systems. Any non-zero value can be assigned to the service name in this section.
- Similarly, the [APP VERSION ROLES] section is used to associate applications and specific versions of applications found in the Add/Remove Programs group. The left value is the application ID string (or sub-string). You should generally NOT include the full version in the text identifier. A version sub-string with Major.minor values is suffixed. The Role ID should match the Service Role ID (where appropriate) and then include a 2 to 3-digit version number. (for example, Exchange is identified in the services via "EXC", this section identifies the specific version – for example – Exchange 2013 would be "EXC13".
- The [IGNORE APP-VER ROLES] section defines any applications that might match but should be excluded.
- Save the file (and return it to the VSA folder if necessary).
- Run the Audit procedure or wait until the next audit completes. Verify that the Service Code appears in the System Roles custom field for the agent where the service exists. Make sure that it isn't on other systems, particularly if you defined an IGNORE SERVICES value.

Create the Custom Monitor Set

Create Service and Event Log monitor sets as you normally would. Be sure to follow the MSP Builder format for defining a Monitor Set ID as described in the Operation section. Note the name of the monitor sets that you create.

Create a View for your Custom Auto-Pilot Policy

- Navigate to Agent / Agents / Manage Agents
- Click the icon to Edit a view

- Select an XAPC view – XAPC_DNS Server is a good choice.
    - Edit the Title and replace "DNS Server" with your custom name. Change the XAPC to ZAPC to identify it as a custom view.
    - Click Save As, remove "Copy of", then click OK.
- Make sure your new view is selected, then click **Define Filter**
- Scroll down to **System Roles** and replace "DNS" with your custom ID. Do the same in the Policy Control field.
- Scroll up, click Apply, then Save the new view.

Create the Auto-Pilot Policy for Custom Monitoring

- Navigate to the Policy Management / Configure / Policies section
- Select the _MSP-Customized \ Custom Monitors (Auto-Pilot) folder.
- Create a new policy using the name "Monitor: (Auto) *service*", using an appropriate service or application name.
- Add the Alerts and Monitor Sets as appropriate
    - If you defined an Event Log monitor, enable Alerts, then use Add Alert to add your custom Event Log monitor.
    - If you defined a Service monitor, enable Monitor Sets, then use Add Monitor Set to add your custom Service monitor.
- Set the view to the ZPOL view you created in the previous step.
- Click **Save and Apply** to complete the new policy setup.

The only remaining step is to link the custom Auto-Pilot policy to the Org Root folder. You can link them individually or via an entire folder. We suggest that you create a folder called "Custom Policies (Auto-Pilot), place all of your custom Auto-Pilot policies there and link the folder directly to the Org Root folder. A second folder – "Custom Policies" – can be used to maintain policies that you link to specific customer groups or individual agents. Keeping your custom objects in separate folder structures ensures that MSP Builder updates won't destroy any custom items.

Collecting Data for Reporting

Many MSPs find the customization of the Daily Audit useful for reporting on items that don't need to be monitored. We provide a separate mechanism for this purpose that will prevent overloading the System Roles field, which could prevent monitors from applying or maintenance tasks from running.

This works the same way as noted above, with three small exceptions:

1. The MSP must create a System Data custom field before making any other changes if it isn't already present. Failure to create this custom field will cause the Daily Audit tool to report failures when this field isn't found, even though all RMM Suite data may have been successfully collected.
2. Create a section called "MSP SERVICE ROLES" in the AuditRoles.ini file and populate it in the same manner as described above for "SERVICE ROLES". This will populate the System Data field with the codes for the services that it finds.
3. Create a section called "MSP APP VERSION ROLES" in the AuditRoles.ini file and populate it in the same manner as described above for "APP VERSION ROLES". This will populate the System Data field with the codes for the applications that it finds.

The ID values used in the MSP's System Data sections do not need to be different from those that populate the System Roles field. You should, however, avoid using the same detections in both sections.

NOTE: *The System Roles and System Data custom field data is available in the SysInfo.ini file and in the registry under* ***HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\RMM****, in the* ***Roles*** *or* ***SystemData*** *values. This may be useful for MSP-created scripts.*

Agent On-Boarding Automation

The Agent Init procedures are designed to run automatically the first time that an agent checks in to VSA. After the basic configuration tasks are performed, the **Init – 3** procedure runs the RMUOBA.BMS app. This app uses a configuration file to invoke a series of agent procedures directly from the agent to perform common and customer-specific configuration tasks. By using a configuration file, the need to update procedures (and the corresponding required experience) is eliminated. The BMS app is deployed and executed automatically – the only configuration needed is in the RMUOBA.INI file.

RMUOBA.INI – Config File

There is a single configuration file that supports the common and all customer-specific tasks. This file is located in the VSASharedFiles folder – specifically in the MSPB_Core\Root\Base_MSP subfolder. Note that if you are utilizing the Tenancy feature, you will need to place a properly customized copy of this configuration file in each tenant's subfolder.

The configuration file defines procedures to run on all agents, exclude from specific orgs, and run only for specific orgs.

All Agents

There are 3 sections that apply to All Agents:

- [ALL] Run these procedures on all agents
- [ALL-WKSTNS] Run these procedures on all workstations
- [ALL-SERVERS] Run these procedures on all servers

In each section, one or more Agent Procedure names can be defined using the following format:

```
Agent Procedure Name=Y|N|ALL
```

The actual procedure name should be defined. A good way to accurately set this is to select the procedure in VSA under Managed Procedures - Schedule/Create and click Rename Procedure. Select the procedure name, copy the text, then click Cancel. Paste the name into the RMUOBI.ini file in the correct location and add the control parameter. The control allows a procedure to be defined but disabled.

Procedures are NOT run on any machine in an unmanaged group by default. You can define specific tasks to run on both managed and unmanaged agents by setting the control parameter to "all".

All Agent Exclusions

There may be times when a procedure that you typically run on all agents needs to be excluded from specific organizations. This might be needed when an alternate procedure is needed. There are three sections related to these exclusions:

- [ALL_EXCLUDE] Exclude the procedure from all org agents
- [ALL-WKSTNS_EXCLUDE] Exclude the procedure from all org workstations
- [ALL-SERVERS_EXCLUDE] Exclude the procedure from all org servers

In each section, one or more Agent Procedure names can be defined, along with a list of Org IDs where the procedure should not be run. Use the following format to define exclusions:

```
Agent Procedure Name=org-1,org-7
```

An agent procedure name can be listed only once per section – any other entry will be ignored! You can specify any number of *unique* agent procedures in each section.

Org-Specific

The org-specific configuration works the same as the All Agents section, except that each section name starts with the VSA Organization ID. This restricts the commands to agents in these organization machine groups.

- [ORG-ALL] Run these procedures on all agents for the specified org
- [ORG-ALL-WKSTNS] Run these procedures on all workstations for the specified org
- [ORG-ALL-SERVERS] Run these procedures on all servers for the specified org

Procedure-Specific Controls

When a procedure is run, it is monitored on 15-second intervals to determine when it has completed. Additional procedures will not run until the previous procedure completes or times-out. If a procedure does not complete within the timeout period, it is marked as having failed.

Procedures that fail will be logged but will not cause the app to fail. Any remaining procedures will be run. If *any* procedure fails to run or complete within the allowed time, the app will report a failure. The Init – 4 procedure is used to report whether the Onboard Automation task was 100% successful.

There are times where a procedure will require more time to complete, or where a procedure must run for later procedures to function. The Procedure-Specific Controls allow these conditions to be accounted for. The following section can be defined for each procedure as needed to adjust the maximum wait time or identify the procedure as required for later procedures.

`[Agent Procedure Name]`	Define the Agent Procedure that uses the special controls
`MaxWait=15`	Define the maximum wait time, in minutes. Cannot be less than 10.
`Required=Y`	Marks the procedure as required. If it fails, all further processing stops.

If the MaxWait value is set to less than 10, the default of 10 minutes will be used. As the app checks every 15 seconds, as soon as the task reports complete the wait-time ends and the next procedure (if any) is executed. If the procedure execution exceeds the MaxWait time, the process is marked as having failed.

Completion

When the app completes processing all specified procedures, it invokes the **ALL-Agent Onboarding - 4 - Agent Onboard Status** procedure. This procedure simply retrieves the status log and determines the PASS/FAIL status of the RMUOBA app. If the log contains “FAIL”, the procedure is forced to fail, providing a clear indication of the onboarding process. If the step 4 procedure fails, the onboarding log file on the agent should be reviewed to determine which procedure failed. Manual remediation may be required for the onboarding process to be considered completed.

Repeat Processing

The RMUOBA app maintains the status of successful procedure execution and will prevent any procedure that ran successfully from running a second time if the RMUOBA app is executed multiple times. If the procedures should be re-run even if originally successful, then a “--FORCE” argument should be used when invoking the app. This will require manually running the app directly on the endpoint to include that argument - from a command prompt, CD to the KWorking folder and run **`bin\RMMKSE.EXE bin\RMUOBA.BMS --FORCE`**.

Should the procedure be run a second time, any onboarding procedure not run previously will be run.

Managed / Unmanaged Agent Processing

By default, the onboard automation is attempted on ALL agents, including those in an unmanaged group, but tasks are suppressed if the agent is in an unmanaged group and the control parameter is anything but “ALL”. To completely suppress the execution of the application on unmanaged agents, the “SkipUnmanaged=Y” argument can be added in the COMMON section of the configuration file. Neither the COMMON section nor the SkipUnmanged value are present in the default file and must be added if needed. The application will still execute the status procedure to notify VSA that it has completed.

Appendix I: Monitor Set Details

This appendix provides a detailed list of Service and Event Log monitoring that is provided with the RMM Suite solution.

Service Monitoring

Monitor Set ID	Event ID	Alert Description	Priority
ADDC.SVC.S.P1.Act	IsmServ kdc Netlogon W32Time	A critical AD svc stopped, multiple restarts failed	1
Altigen.SVC.S.P3.Act	AGJServ AltiBack AltiCTProxy AltiExchIntg AltiFTPUploader AltiGateway AltiGen External Features Server AltiGen TAPI Proxy Server AltiKeep AltiLogger AltiMA AltiPhoneServ AltiPop3 AltiServ AltiSmtp AltiVMServ AUService GphoneService OpenLDAP-slapd postman	An Altigen svc stopped, multiple restarts failed	3
BBS.SVC.S.P2.Act	BBAttachServer BlackBerry Controller BlackBerry Database Consistency BlackBerry Dispatcher BlackBerry MDS Connection BlackBerry Policy Service BlackBerry Router BlackBerry Server Alert BlackBerry SyncServer	A critical Blackberry svc stopped, multiple restarts failed	2

Core.SVC.S.P2.Act	BFE ClusSvc CryptSvc DcomLaunch Dhcp Dnscache Eventlog EventSystem Iphlpsvc Lanmanserver Lanmanworkstation LmHosts MpsSvc Netlogon Netprofm NlaSvc PlugPlay ProfSvc RpcEptMapper RpcSs SamSs Schedule SENS SessionEnv Spooler TermService TrkWks UmRdpService W32Time winmgmt	A critical core svc stopped, multiple restarts failed	2
Core.SVC.W.P3.Act	AudioEndpointBuilder Audiosrv BFE CryptSvc DcomLaunch` Dhcp DPS Eventlog EventSystem Iphlpsvc Lanmanserver Lanmanworkstation MpsSvc Netlogon Netprofm NlaSvc PcaSvc PlugPlay Power ProfSvc RpcEptMapper RpcSs SamSs Schedule SENS SessionEnv Spooler TermService TrkWks UmRdpService W32Time winmgmt	A core svc stopped, multiple restarts failed	3

Core-ND.SVC.W.P3.Act	AudioEndpointBuilder Audiosrv BFE CryptSvc DcomLaunch` Dhcp DPS Eventlog EventSystem Iphlpsvc Lanmanserver Lanmanworkstation MpsSvc Netprofm NlaSvc PcaSvc PlugPlay Power ProfSvc RpcEptMapper RpcSs SamSs Schedule SENS SessionEnv Spooler TermService TrkWks UmRdpService W32Time winmgmt	A core svc stopped, multiple restarts failed	3
DHCPSvr.SVC.S.P1.Act	DHCPServer	A critical DHCP svc stopped, multiple restarts failed	1
DNS.SVC.S.P1.Act	DNS	A critical DNS svc stopped, multiple restarts failed	1
Exchange.SVC.S.P1.Act	ADAM_MSExchange EdgeCredentialSvc IMAP4Svc MSExchangeADTopology MSExchangeAntispamUpdate MSExchangeEdgeSync MSExchangeFBA MSExchangeImap4 MSExchangeIS MSExchangeMailboxAssistants MSExchangeMailboxReplication MSExchangeMonitoring MSExchangeMTA MSExchangePop3 MSExchangeProtectedServiceHost MSExchangeRepl MSExchangeRPC MSExchangeSA MSExchangeServiceHost MSExchangeSubmission MSExchangeTransport MSExchangeTransportLogSearch Pop3Svc SMTPSVC	A critical Exchange svc stopped, multiple restarts failed	1

FileSvr.SVC.S.P2.Act	Dfs DFSR	A critical File Server svc stopped, multiple restarts failed	2
GFI.SVC.S.P1.Act	MARCore MarIMAP MARMAIS MARSearch MARStore MARVSS	A critical GFI svc stopped, multiple restarts failed	1
HMail.SVC.S.P3.Act	hMailServer	An HMail svc stopped, multiple restarts failed	3
IIS.SVC.S.P1.Act	AppHostSvc FTPSVC IISADMIN MSFtpsvc W3svc WAS WMSVC	A critical IIS svc stopped, multiple restarts failed	1
RDSCB.SVC.S.P1.Act	TScPubRPC Tssdis	A critical RDSBroker svc stopped, multiple restarts failed	1
RDSGW.SVC.S.P2.Act	TSGateway	A critical RDS GW svc stopped, multiple restarts failed	2
RDSLS.SVC.S.P3.Act	TermServLicensing	A critical RDS License svc stopped, multiple restarts failed	3
ShadowProt.SVC.S.P3.Act	ShadowControl ImageManager ShadowProtectSvc	A ShadowProtect svc stopped, multiple restarts failed	3
SharePoint.SVC.S.P1.Act	MSSEARCH SPAdmin SPTimer SPTrace	A critical SharePoint svc stopped, multiple restarts failed	1
SMTP.SVC.S.P2.Act	SMTPSVC	A critical SMTP svc stopped, multiple restarts failed	2
SQL.SVC.S.P1.Act	MSDTC MSSQLSERVER MySql ReportServer SQLSERVERAGENT	A critical SQL svc stopped, multiple restarts failed	1
vcenter.SVC.S.P1.Act	ADAM_VMwareVCMSDS vCOConfiguration Vctomcat vimPBSM vimQueryService VMUSBArbService VMWareCertificateService VMwareDirectoryService VMwareIdentityMgmtService VMwareKdcService Vmwarelogbrowser VMwareOrchestrator VMwareSTS Vpxd vspherewebclientsvc	A critical vCenter svc stopped, multiple restarts failed	1

Veeam.SVC.S.P3.Act	MSSQL$VEEAM Veeam Backup and Replication Veeam Backup Catalog Data VeeamAgentRoutingSvc VeeamCloudSvc VeeamDeploymentService VeeamNFSSvc VeeamTransportSvc VeeamVssSupport	A Veeam svc has stopped, multiple restarts failed	3

Event Log Monitoring

Monitor Set ID	Event ID	Alert Description	Priority
MB-ADDC.EVT.S.P2.Alm	1150	AD Svcs-CRITICAL: Schema modification attempt failed	2
	1311	AD Svcs-CRITICAL: KCC reports directory partition problems	
MB-ADDC.EVT.S.P3.Alm	1008	AD Svcs: KCC failed to initialize.	3
	1016	AD Svcs: The schema could not be loaded	
	1084	AD Svcs: No preferred bridgehead servers can replicate	
	1126	AD Svcs: Unable to establish connection with global catalog	
	1136	AD Svcs: Schema index create failed for specified attribute	
	1312	AD Svcs: KCC Replication Path Computation error	
	1411	AD Svcs: Mutual authentication SPN failed for a DC	
	1473	AD Svcs: Intersite Messaging svc stopped	
	1539	AD Svcs: Could not disable software-based disk write cache	
	1546	AD Svcs: Schema attribute conflict found during replication	
	1567	AD Svcs: Failed to replicate the directory partition	
	1844	AD Svcs: Failed to discover replication partners	
	1865	AD Svcs: KCC unable to form a spanning tree network topology	
	1925	AD Svcs: Failed to establish a replication link	
	1964	AD Svcs: Directory service denied a replication attempt	
	1977	AD Svcs: Directory service denied a replication attempt	
	2087	AD Svcs: DNS failed to resolve the source DC to IP address	
	2088	AD Svcs: Could not use DNS to resolve the IP of source DC	
	2089	AD Svcs: Directory partition exceeded backup latency period	
	9	AD Svcs: Policy module load fail. Cert Service did not start	
MB-ADLK-C.EVT.S.P3.Alm	4740	AD User Lock-out	3
MB-AINS-A.EVT.S.P2.Alm	100	Alert Caller-CRITICAL: Check Failed	2
	120	Alert Caller-CRITICAL: Call Exception	
MB-CORE-S.EVT.S.P3.Alm	333	Core Monitors: A Registry I/O Operation Failed Unrecoverably	3
	2020	Core Monitors: Allocation from the System Paged Pool failed	
	8	Core Monitors: Release Writes Command timed out	
MB-CORE.EVT.S.P2.Alm	401	Task Scheduler-CRITICAL: Service failed to start	2
	412	Task Scheduler-CRITICAL: Svc failed to launch startup tasks	
	3002	Windows Init-CRITICAL:	
	2001	Server Svc-CRITICAL: Server service not started.	
	2505	Server Svc-CRITICAL: Duplicate name on network.	

Monitor Set ID	Event ID	Alert Description	Priority
MB-CORE.EVT.S.P3.Alm	1002	Group Policy: Processing failed-system allocation failure	3
	1096	Group Policy: Processing failed to apply registry settings	
	1125	Group Policy: Processing failed-internal system error	
	1126	Group Policy: Clock not synchronized with DC	
	1127	Group Policy: Processing failed-internal system error	
	1130	Group Policy: Bad GPO Name, path, or script name.	
	311	Task Scheduler: Task engine could not start-bad command	
	403	Task Scheduler: Remote server returned error when running job	
	1015	Windows Init -	
	4102	WinLogin - Object at the specified path does not exist	
	4103	WinLogin -	
	1123	Server Agent: POST errors detected	
	7003	Svc Control Manager: Dervice depends on nonexistent service	
	7016	Svc Control Manager: Service reported an invalid state	
	7022	Svc Control Manager: DTC service hung on start	
	7023	Svc Control Manager: Service reported error and terminated	
	7024	Svc Control Manager: Service reported error and terminated	
	7033	Svc Control Manager: SCM did not initialize-restarting	
	7034	Svc Control Manager: SCM terminated unexpectedly	
	7038	Svc Control Manager: Invalid Service Account	
	7041	Svc Control Manager: Invalid Service Account Group	
MB-CORE.EVT.W.P3.Alm	1002	Group Policy: Processing failed-system allocation failure	3
	1096	Group Policy: Could not apply registry-based settings	
	1125	Group Policy: Processing failed-internal system error	
	1126	Group Policy: Clock not synchronized with DC	
	1127	Group Policy: Processing failed-internal system error	
	1130	Group Policy: Bad GPO Name, File system path, or script name	
	311	Task Scheduler: Process could not start-error in command	
	401	Task Scheduler: Remote server returned an error running job	
	403	Windows Init -	
	412	WinLogin - Object at the specified path does not exist	
	55	WinLogin -	
	2001	Server Agent: POST errors detected.	
	2505	Svc Control Mgr: Service dependent upon nonexistent service	
	7016	Svc Control Mgr: Service reported an invalid current state	
	7022	Svc Control Mgr: DTC service hung on start	
	7023	Svc Control Mgr: Service reported an error and terminated	
	7024	Svc Control Mgr: Service reported an error and terminated	
	7033	Svc Control Mgr: SCM did not init. System is restarting	
	7034	Svc Control Mgr: SCM terminated unexpectedly.	
	7038	Svc Control Mgr: Service start fail-bad credentials	
	7041	Svc Control Mgr: Access to managed service denied	
MB-CRSH-S.EVT.S.P3.Alm	41	Crash Detection: Unexpected System Restart occurred	3
MB-DHCP-S.EVT. S.P2.Alm	1008	DHCP Server-CRITICAL: Service failure-forced shutdown	2

Monitor Set ID	Event ID	Alert Description	Priority
MB-DHCP-S.EVT.S.P3.Alm	1001	DHCP Server: Computer was not assigned an address.	3
	1002	DHCP Server: Computer IP lease has been denied by the server	
	1003	DHCP Server: Unable to renew assigned address	
	1004	DHCP Server: Service failed to initialize the database	
	1005	DHCP Server: Service failed to initialize Winsock	
	1006	DHCP Server: Service failed to initialize as RPC server.	
	1007	DHCP Server: failed to deliver address. Auto-configured IP	
	1020	DHCP Server: Scope has no addresses available	
	1034	DHCP Server: Service failed to init one or more DLLs	
	1045	DHCP Server: Service failed to init. Domain not authorized	
	1046	DHCP Server: Service failed to init. Domain not authorized	
	1050	DHCP Server: Service encountered a network error	
	1058	DHCP Server: Service failed to initialize config parameters	
	1059	DHCP Server: Failed to find authoritative directory server	
	1063	DHCP Server: Scope has no addresses available.	
	1102	DHCP Server: Service authorization failed.	
	1104	DHCP Server: Service authorization failed.	
	1144	DHCP Server: Service computer IP assigned by DHCP.	
MB-DNS.EVT.EVT.S.P2.Alm	10	DNS Server-CRITICAL: Can't start-NTDS service not started.	2
	111	DNS Server-CRITICAL: Can't create thread-low resources	
MB-DNS.EVT.EVT.S.P3.Alm	140	DNS Server: Unable to initialize RPC.	3
	500	DNS Server: Invalid or corrupted registry data	
	505	DNS Server: Invalid or corrupted registry data	
	707	DNS Server: Server not root authoritative. Invalid Cache	
	1003	DNS Server: Specified hostname could not be found	
	1501	DNS Server: Unable to parse a database file	
	3151	DNS Server: Unable to write zone file	
	7502	DNS Server: Unable to service request-memory shortage	
MB-EXC.EVT.S.P3.Alm	24	Exchange Service: Web Services using an expired certificate	3
	12015	Exchange Service: Web Services using an expired certificate	
	12017	Exchange Service: Web Services certificate due to expire	
MB-HyperV-S.EVT.S.P2.Alm	4010	Hyper-V-CRITICAL: Version mismatch reported	2
	16050	Hyper-V-CRITICAL: Disk space	
	6060	Hyper-V-CRITICAL: Environment Faults	
	16210	Hyper-V-CRITICAL: Disk merge failed to merge disk on delete	
	19090	Hyper-V-CRITICAL: Disk merge has been interrupted.	
	19100	Hyper-V-CRITICAL: Disk merge failed to complete.	
	3040	Hyper-V-CRITICAL: Initialization failed.	
MB-HyperV-S.EVT.S.P3.Alm	4096	Hyper-V: Configuration inaccessible. Bad path	3
	12570	Hyper-V: Init failed. Virt net adapter-unreachable network	
	12572	Hyper-V: Dynamic MAC addr not assigned. No net connectivity	
	12140	Hyper-V: Failed to open attachment.	
	12240	Hyper-V: Failed to open attachment. Attachment not found	
	12290	Hyper-V: Failed to open attachment. Access Denied	
	10104	Hyper-V: Failed to revert VSS snapshot on one or more VHDs	
	12240	Hyper-V: Failed to open attachment. Attachment error	
	12290	Hyper-V: Failed to open attachment. Access Denied	
	16020	Hyper-V: Network resource is no longer available	
	16060	Hyper-V: Disk space caused operations to pause	
	16310	Hyper-V: Configuration inaccessible because it is corrupt	
	16400	Hyper-V: Environment Faults	
	16410	Hyper-V: Configuration inaccessible. Bad path specified	
	16420	Hyper-V: Unable to access data folder of the snapshot	
	3030	Hyper-V: Insufficient memory or processing capacity	
	3050	Hyper-V: Component that raised this event is corrupt	
	3080	Hyper-V: Failed to create or access saved state file	
	3122	Hyper-V: Environment Faults	
	3320	Hyper-V: Failed to create memory contents of file	
	12140	Hyper-V: Failed to open attachment	

Monitor Set ID	Event ID	Alert Description	Priority
MB-HyperVRepl-S.EVT.S.P3.Alm	19060 20880 29012 29292 29296 32022 32026 32032 32088 32315 32326 32346 32350 32366 32510 32546 32572 32587 32592 33676 33680 33812 33824 33826	Hyper-V: Repl Failure-Operation already in progress Hyper-V: Repl Failure-Failed to delete folder Hyper-V: Repl Failure-Failed to apply replicated changes Hyper-V: Repl Failure-Operation timed out Hyper-V: Repl Failure-Unexpected network error Hyper-V: Repl Failure- Hyper-V: Repl Failure-Failed to generate VM delta Hyper-V: Repl Failure-Insufficient disk space Hyper-V: Repl Failure-Replication suspended on Replica host Hyper-V: Repl Failure-Failed to replicate VM changes Hyper-V: Repl Failure-Requires resync Hyper-V: Repl Failure-Insufficient disk space. Hyper-V: Repl Failure-Requires resync, tracking error Hyper-V: Repl Failure-Failed to apply the log file Hyper-V: Repl Failure-Failed to delete the log file. Hyper-V: Repl Failure-Operation cannot be performed Hyper-V: Repl Failure-Failed to resynchronize changes Hyper-V: Repl Failure-Failed to update log file time Hyper-V: Repl Failure-Failed to apply changes Hyper-V: Repl Failure-Insufficient disk space Hyper-V: Repl Failure-Operation failed Hyper-V: Repl Failure-Reference point export failed Hyper-V: Repl Failure-Insufficient disk space Hyper-V: Repl Failure-Insufficient disk space	3
MB-IIS.EVT.S.P3.Alm	1003 1004 1007	IIS Service: Site disabled-URL prefix not registered by WWW IIS Service: Site disabled-RL prefix not registered by WWW IIS Service: Site disabled-Network binding may be in use	3
MB-KAVL.EVT.W.P3.Alm	4660	KAV Licensing: License is invalid or has expired.	3
MB-KNMG-A.EVT.S.P3.Act	0	Kaseya Network Monitor: Gateway Service Failure	3
MB-LDS-S.EVT.S.P3.Alm	24595 24596 2048 2057 2094 1200 1216	Local Disk Storage: Drive FAILED. Local Disk Storage: Drive failure predicted. Local Disk Storage: Device FAILED. Local Disk Storage: Virtual Disk Degraded (RAID) Local Disk Storage: Drive failure predicted. Local Disk Storage: Drive Status Changed Local Disk Storage: Drive Status Changed	3
MB-LDS-S.EVT.W.P2.Alm	7206	Local Disk Storage: Failure reported	2

Monitor Set ID	Event ID	Alert Description	Priority
MB-NTS.EVT.S.P3.Alm	1	Network Time Service:	3
	4	Network Time Service:	
	11	Time Service: Can't use domain time - not domain member	
	12	Time Service: PDCe - Unable to use domain time source	
	15	Time Service: Unable to use domain time source. No DC found	
	21	Time Service: Sync failed-no available configured provider	
	28	Time Service: Sync failed-no configured provider reachable	
	30	Time Service: Can't start-Error reading registry data	
	31	Time Service: System time zone information is corrupt	
	32	Time Service: System time zone information is corrupt	
	34	Time Service: Sync failed. Time difference not corrected	
	36	Time Service: Snyc failed. No usable time from providers	
	39	Time Service: Sync failed-Failed to register config change	
	41	Time Service: Sync failed. No configured provider is running	
	42	Time Service-CRITICAL: Event queued creation already opened	
	44	Time Service: Sync failed-NTP provider failed and shut down	
	45	Time Service: Sync failed-NTP provider failed and shut down	
	46	Time Service: Sync failed-NTP provider failed and shut down	
	50	Time Service: Sync broken-Time difference for at least 2s	
	130	Time Service: Sync failed-NTP unable to set domain peer	
	131	Time Service: Sync failed-NTP unable to set domain peer	
	141	Time Service: Sync failed-No configured provider is running	
	142	Time Service: Sync failed-Local clock inaccurate-svc stopped	
	144	Time Service: Sync failed-Not advertising as time source	
MB-PSP.EVT.S.P2.Alm	363	Spooler Service-CRITICAL: Failed to start	2
MB-PSP.EVT.S.P3.Alm	99	Spooler Service-CRITICAL: Fatal error in critical operation	3
	315	Spooler Service: Failed to share printer	
	319	Spooler Service:	
	361	Spooler Service:	
MB-RDCM.EVT.S.P3.Alm	1149	RDS Connection Manager Service:	3
MB-RDGW.EVT.S.P3.Alm	203	RDS Gateway Service: Exceed maximum allowed sessions	3
MB-RDLC.EVT.S.P3.Alm	12	RDS Licensing Service:	3
	22	RDS Licensing Service:	
	36	RDS Licensing Service:	
MB-SQL.EVT.S.P3.Alm	3041	SQL Server: Failed command BACKUP DATABASE master	3
	208	SQL Server: Invalid object name. Database content error	
	12291	SQL Server: Message timed out. Time used > time to live	

Custom Event Alerts for MSP Builder Maintenance and Smart Monitoring

Monitor Set	Event ID	Alert Description	Priority
MB-ASM-A.EVT.S.P3.Alm	101	ASM: Missing Configuration File	3
	102	ASM: No User Logged In	
	105	ASM: Multiple Start Failures	
MB-K32-A.EVT.W.P5.Chk	1	BMS Script Failure: See content for details	3
MB-ICC-A.EVT.S.P3.Alm	131	Internet Connection Monitor: On Primary Service	3
	132	Internet Connection Monitor: On Backup Service	
	135	Internet Connection Monitor: Not Configured	
	136	Internet Connection Monitor: Deploy to Unsupported Platform	
MB-MNT-A.EVT.S.P3.Alm	102	Daily Maint-Server: Invalid configuration	3
	104	Daily Maint-Server: Task Failed to Run	
	108	Daily Maint-Server: Excessive Uptime	
	131	Daily Maint-Server: Defrag Failed	
	191	Daily Maint-Server: ChkDsk Alert-Repair Scheduled	
	192	Daily Maint-Server: ChkDsk Alert-Errors Reported Post-Repair	
	193	Daily Maint-Server: ChkDsk Alert-SMART Error Reported	
MB-MNT-A.EVT.W.P3.Alm	102	Daily Maint-Wkstn: Invalid Configuration	3
	104	Daily Maint-Wkstn: Task Failed to Run	
	108	Daily Maint-Wkstn: Excessive Uptime	
	109	Daily Maint-Wkstn: System Tray Utility Failure	
	13	Daily Maint-Wkstn: Defrag Failed	
	151	Daily Maint-Wkstn: Local Backup Process Failed	
	152	Daily Maint-Wkstn: System Restore Point Failures	
	191	Daily Maint-Wkstn: ChkDsk Alert-Repair Scheduled	
	192	Daily Maint-Wkstn: ChkDsk Alert-Errors Reported After Repair	
	193	Daily Maint-Wkstn: ChkDsk Alert-SMART Error Reported	
MB-SM-A.EVT.S.P3.Alm	121	Smart Monitor-Server: IN SAFE MODE	3
	122	Smart Monitor-Server: IN DSR MODE	
	123	Smart Monitor-Server: Safe Mode Enabled at Next Boot	
	126	Smart Monitor-Server: Boot during Business Hours	
	129	Smart Monitor-Server: Can't determine uptime	
	161	Smart Monitor-Server: Disk Capacity Trending Notice	
	162	Smart Monitor-Server: Low Disk Capacity	
	163	Smart Monitor-Server: Low Disk Capacity - CRITICAL	
	165	Smart Monitor-Server: Disk Capacity Monitors are suspended	
	169	Smart Monitor-Server: Invalid custom parameter	
	181	Smart Monitor-Server: Domain Time not in sync < 4m	
	181	Smart Monitor-Server: Domain Time not in sync > 4m-CRITICAL	
	182	Smart Monitor-Server: Time Sync Config Error	
	184	Smart Monitor-Server: Multiple NTP Resync Failures	
	187	Smart Monitor-Server: W32Time - Service Missing - CRITICAL	
MB-SM-A.EVT.W.P3.Alm	111	Smart Monitor-Wkstn: Antivirus - No AV Product Detected	3
	112	Smart Monitor-Wkstn: Antivirus - Definitions are Outdated	
	113	Smart Monitor-Wkstn: Antivirus - Not Running	
	114	Smart Monitor-Wkstn: Antivirus - Multiple Products Running	
	115	Smart Monitor-Wkstn: Antivirus - Protection is Disabled	
	116	Smart Monitor-Wkstn: Antivirus - Not Preferred Product	
	161	Smart Monitor-Wkstn: Disk Capacity Trending Notice	
	162	Smart Monitor-Wkstn: Low Disk Capacity	
	163	Smart Monitor-Wkstn: Low Disk Capacity - CRITICAL	
	169	Smart Monitor-Wkstn: Invalid custom parameter	
	181	Smart Monitor-Wkstn: Domain Time not in sync > 4m-CRITICAL	
	182	Smart Monitor-Wkstn: Time Sync Config Error	
	183	Smart Monitor-Wkstn: Alert - NTPDATE.EXE is not present	
	184	Smart Monitor-Wkstn: Alert - Multiple NTP Time Resync actions	
	187	Smart Monitor-Wkstn: W32Time Service Missing-CRITICAL	
MB-SM-TIME-A.EVT.S.P3.Act	185	Smart Monitor-Server: PDCe not configured to use NTP	3
	186	Smart Monitor-Server: Member Server not using NT5DS	

Monitor Set ID	Event ID	Alert Description	Priority
MB-WAE-A.EVT.X.P3.Alm	401 402 403 404 405 412	RMM License Authorization: MSP is Not Licensed RMM License Authorization: Unknown Error RMM License Authorization: Duplicate Machine GUID RMM License Authorization: Auth Rejected RMM License Authorization: Auth Website unavailable RMM License Authorization: Failed to Generate Checksum	3