MSP BUILDER

Tools for MSP Success

RMM Suite for Kaseya VSA
Operations & Customization Guide

Enhanced Maintenance & Monitoring (EMM)

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

Last Updated: 2022/04/02

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The MSP Builder RMM Suite contains proprietary software, including unpublished source code.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 ("COMPANY") CONFIDENTIAL

NOTICE: All information contained herein is and remains the property of COMPANY. The intellectual and technical concepts contained herein are proprietary to COMPANY and may be covered by U.S. and Foreign Patents, patents in process, and are protected by trade secret or copyright law.

Dissemination of this information or reproduction of this material is strictly forbidden unless prior written permission is obtained from COMPANY. Access to the source code contained herein is hereby forbidden to anyone except current COMPANY employees, managers or contractors who have executed Confidentiality and Non-disclosure agreements explicitly covering such access.

The copyright notice above does not evidence any actual or intended publication or disclosure of this source code, which includes information that is confidential and/or proprietary, and is a trade secret, of COMPANY. ANY REPRODUCTION, MODIFICATION, DISTRIBUTION, PUBLIC PERFORMANCE, OR PUBLIC DISPLAY OF OR THROUGH USE OF THIS SOURCE CODE WITHOUT THE EXPRESS WRITTEN CONSENT OF COMPANY IS STRICTLY PROHIBITED, AND IN VIOLATION OF APPLICABLE LAWS AND INTERNATIONAL TREATIES. THE RECEIPT OR POSSESSION OF THIS SOURCE CODE AND/OR RELATED INFORMATION DOES NOT CONVEY OR IMPLY ANY RIGHTS TO REPRODUCE, DISCLOSE OR DISTRIBUTE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING THAT IT MAY DESCRIBE, IN WHOLE OR IN PART.

MSP BUILDER

MSP Builder, LLC
385 Falmouth Ave
Elmwood Park, NJ 07407
201-796-0404

Contents

Introduction

The MSP Builder Endpoint Maintenance & Monitoring (EMM) expands the RMM Suite with a highly configurable set of tools that perform daily endpoint maintenance and implements Smart-Monitor technology for advanced alerting. These components can be deployed selectively to agents as a value-added service offering. Maintenance and Smart Monitors are independent of each other, allowing you to deploy none, either, or both components.

Overview

The EMM Suite consists of the following standard components.

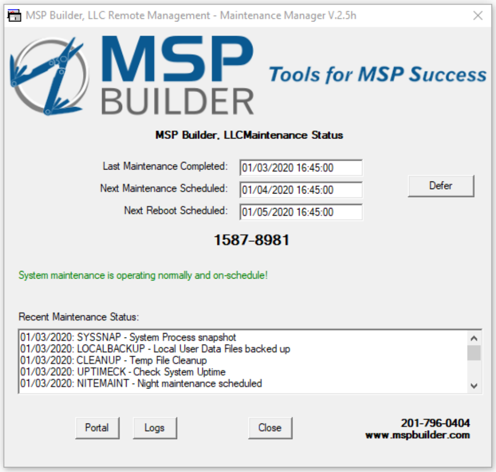
Maintenance Suite

The EMM Maintenance Suite is a combination of procedures, scripts, and applications that combine to autonomously perform daily maintenance tasks. These ensure the ongoing well-being of the computers by automating many administrative tasks. Maintenance runs in two passes, with system checks and data collection tasks performed during the day and more invasive tasks scheduled for after business hours. The standard tools perform the most needed ongoing maintenance for servers and workstations, such as disk cleanup, disk defrag (SSD aware!), log management, and Time Sync checks. Custom tasks are easily added to meet any MSP or customer requirements.

A secondary suite provides tools for interactive use when diagnosing or remediating issues. MSP-specific tools can also be delivered through our standard tool deployment process to ensure that your favorite support tools are always available. The tool folder is automatically added to the System PATH to make them available from Start/Run or the command prompt.

User Interface

The Maintenance Suite includes a user interface that allows the maintenance tasks to communicate with the logged-on user. It can provide the status of maintenance, alerting when maintenance has not completed, and even remind the user of upcoming patch or software update schedules. The interface loads when the user logs on, displays briefly to show the recent status, and then minimizes to the system tray. RMM procedures are provided to send messages to the user and get basic responses (yes/no, OK) from the user, which can be used in custom procedures. (This replicates built-in Kaseya functions, but the prompt will display the MSP logo and match the other RMM Suite message display style.)


MSP Builder, LLC Remote Management - Maintenance Manager V.2.5h

MSP Builder, LLCMaintenance Status

Last Maintenance Completed: 01/03/2020 16:45:00
Next Maintenance Scheduled: 01/04/2020 16:45:00
Next Reboot Scheduled: 01/05/2020 16:45:00

1587-8981

System maintenance is operating normally and on-schedule!

Recent Maintenance Status:

```
01/03/2020: SYSSNAP - System Process snapshot
01/03/2020: LOCALBACKUP - Local User Data Files backed up
01/03/2020: CLEANUP - Temp File Cleanup
01/03/2020: UPTIMECK - Check System Uptime
01/03/2020: NITEMAINT - Night maintenance scheduled
```

201-796-0404
www.mspbuilder.com

This user interface is fully customizable with the MSP's logo, URL, contact info, and links to your web site and portal. The interface can be suppressed to only display messages and warnings, eliminating the brief daily status display.

Dates will be displayed based on the user's regional settings.

A security code is displayed in the center of the interface. This code is specific to the agent where the interface is running and changes every 60-seconds. The MSP can deploy a complementary tool to their support technicians that can generate the same code when the agent's name is entered. This code can be used to help secure a voice session initiated by the MSP to the end-user by identifying the security code when the call is initiated and having the end user launch the interface and verify the code.

Smart Monitors

As the name implies, these monitors have built-in intelligence to work more efficiently. Smart Monitors feature:

- Transient Suppression Technology – An alert is not triggered unless the event is critical, or a non-critical alert state exists for 3 days. This minimizes alerts from temporary conditions.
- Self-Adjusting Thresholds – alert thresholds are calculated based on the actual local device configuration, eliminating "one size hardly fits any" or "every alert is manually adjusted".
- Self-Remediation – When the alert is first detected, a remediation process is attempted to eliminate the condition reducing alerts and manual intervention for many events.

The Disk Capacity Smart Monitor is licensed as part of Core Automation and does not incur additional cost to deploy.

System Policies

System Policies are used to deploy daily tasks such as maintenance and Smart Monitors and apply the monitors needed by these tools.

Monitor Sets

Monitors related to the Maintenance and Smart Monitor tools are applied by system policy. Views control these based on per-agent configuration settings.

Views

Views are provided to control the deployment of System Policies for Maintenance and Smart Monitors.

Procedure Library

Procedures deploy configuration files and execute the Maintenance and Smart Monitor executables.

Pre-Requisites

The EMM Suite depends upon the MSP Builder Core Automation Suite technology and the methodologies recommended therein. Some of the EMM monitors and procedures may need to be modified if the recommended configuration is not followed.

This document will assume that you are familiar with the RMM Suite and its use and administration. You may find it useful to review that document before implementing the EMM components, or to refer to it to clarify certain operations that are defined in that guide.

Global Configuration Settings

The EMM components of Daily Maintenance and Smart Monitors are generally deployed to all managed customer endpoints. These improve the quality of monitoring, improve endpoint operation, and virtually eliminate false alert events. MSPs are encouraged to consider these "premium" services and charge an additional fee when these are deployed. To help support this model, EMM can be enabled or disabled at various levels.

The **ALL-Set Common Data** procedure controls the global use of the EMM components. Set the GLOBAL:XEMS value to "0" to globally disable EMM tools, or set it to "1" to globally enable use of the EMM Suite. This is a global setting that controls deploying the Maintenance and Smart Monitor procedures to all managed customer organizations. Organizations with a root group of "unm", agents placed into the Audit customer, or in the Special sub-group are considered unmanaged and no monitoring, maintenance, or updating is performed on these agents. Only auditing and patch scanning are performed to allow status reporting.

Set the Managed Variable **DisableEMM** to 1 to turn EMM off on a customer-org or group-specific manner. This setting requires that the GLOBAL:XEMS value be set to 1 to globally enable EMM services. This is the recommended method to control deployment when offered as a premium service.

The Daily Maintenance and Smart Monitor deployments can also be individually disabled on a per-agent basis by deploying the appropriate MAINT or SMON Policy Blocker. This adds these terms to the Policy Control custom field, disabling the corresponding component. This is generally used to temporarily suspend these services when a system is being deployed or upgraded and any changes should be avoided.

Note that when either the global or org-specific control is set to disable EMM, the User Interface will still be displayed in a simplified format to allow use of the MFA security code. To fully disable the User Interface, the Managed Variable “MGUIArgs” must be set to “--X”.

Operational Overview

This chapter will provide information on the day to day operation of the EMM components. Some information will be in summary format, with detail provided in a separate chapter.

The automation process starts with System Policies being linked to the organizational root. These are "Auto-Pilot" policies that will apply to all systems unless an exclusion is detected. There are several standard exclusions, which are the same as those used throughout the RMM Suite products.

- If the machine group contains ".unm.", indicating an *unmanaged* customer, or a managed customer that is still in an unmanaged state, not yet fully on-boarded.
- If the machine group contains ".special", indicating a device requiring special handling.

N.B. "unm" is intended to be used as the "root" group name and expect that a sub-group will be created below them. This is to ensure that customers with identities that end with these text parts will not be detected. Any agent placed directly into the "unm" folder will **NOT** be processed as expected, and all automation will be enabled!

Daily Tasks

A pair of procedures are used to deploy the daily tasks – one for Servers and the other for Workstations. Both procedures are similar, except that they identify the target of "server" or "workstation". The major difference that results is the set of configuration files that are deployed. There are separate files with different tasks and settings for servers and workstations. One key difference is that server tasks are scheduled for outside of business-hours to minimize any possible impact, while workstation tasks are scheduled for times that the system is most likely to be available. Another small difference is that the workstation tasks will recycle the end user interface, ensuring that it is updated.

Maintenance

The Daily Tasks procedure will launch the RMM_Maintenance procedure if maintenance is not suppressed via Policy Control. (see below for more information on Policy Control.)

For all computers, the configuration files are first deleted and re-deployed before the maintenance sequence engine is invoked. This ensures that locally modified versions of the configuration file are not used when run with elevated rights.

For workstations, the user interface is also stopped and restarted prior to launching the sequence engine. This ensures that the latest interface tool is running and that it displays briefly to remind the user of upcoming maintenance tasks, reboots, and general status. This brief display can be suppressed through a Managed Variable setting.

Smart Monitors

The Smart Monitor procedure will be run if it is not suppressed via Policy Control. (see below for more information on Policy Control.) The same Smart Monitor procedure is run on servers and workstations – the Smart Monitors will exit without error if the platform they run on is not supported. For example, Server Boot Monitor, which detects business-hours reboots and Safe Mode, will not operate on workstations.

Daily Audit

The daily audit tool runs on every agent without exception unless the DAILY Policy Control value is defined (which prevents the daily procedure from running at all). This ensures data is collected for all agents, regardless of management status.

Policy Control

Policy Control is a text string that contains a series of values that can either block the application of Auto-Pilot policies or the execution of certain procedures such as Smart Monitors and Maintenance. The Policy Control data is stored in the registry of the local computer and replicated back to a Custom data field for each agent by the daily audit and by the procedures that update these values. *It is critical to use the procedures when updating these values, as they sync the system registry and custom field data!*

Policy Control values are always enclosed in dashes. This ensures that values will not be found inside of other values. For example, there could be two components – SQL and SQLX. Without the "-" delimiters, "SQL" would match both values. The procedures ensure that these delimiters are always used.

These are the standard Policy Control values and their meanings:

• BLMON	Excludes the Baseline Monitors.
• EXMON	Excludes all Extended Monitors (role, feature, and application).
• DAILY	Excludes the Daily Tasks procedure (including Audit).
• MAINT	Excludes the Maintenance procedure.
• SMON	Excludes the Smart Monitors from running.
• PATCH	Excludes the agent from participating in patching (Patch Mgmt & Software Mgmt).
• APPUD	Excludes application updates (Via Ninite-based procedures) from running.

Other policy controls can be defined based on the System Role identity that is defined via the daily Audit process. For example, if SQL Server is found on an agent, "SQL" is added to the System Roles custom field, which causes the SQL monitor sets to be applied. Use the **Policy Blocker – Add – Specify** procedure to add "SQL" to the Policy Control field to prevent the SQL monitor sets from being applied.

It is recommended that these policy blockers be used only after careful consideration, as they can prevent monitoring, maintenance, patching/updating, and auditing tasks from being performed. A common need for this is when one machine at a client is not monitored or maintained, or when a particular service or system role should be excluded from monitoring on a development system, but other monitors and management features are required.

Per Agent or Per Group

Policy Control settings are applied to individual agents. When EMM should be deployed selectively to customer organizations or sites, the Managed Variable "DisableEMM" should be used instead of individual Policy Blockers to control where EMM components are used.

Per Smart Monitor

Individual Smart Monitors can easily be suppressed on specific agents by applying the following policy control values in the same method as described in the Policy Control section above:

• SMSSC	Suppress the System Security (AntiVirus) Checks
• SMDCC	Suppress the Disk Capacity Checks
• SMDCT	Suppress only the Disk Capacity Trending alerts (trending is still logged)
• SMSBM	Suppress the Server Boot Monitor checks
• SMNTP	Suppress the Network Time Protocol checks
• SMUSC	Suppress the User Account Security Checks

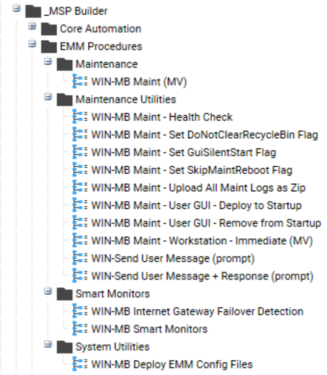
Agent Procedures

The EMM Suite adds several procedures that are related to Maintenance and Smart Monitors, along with the related executables. The procedures ensure that the most current tools and configuration files are deployed just prior to execution.

Overview

Following the standard practices guidelines set in the RMM Suite, the Agent Procedures in the EMM Suite all share several best-practice methods. All EMM tools are found in the **Endpoint M&M** subfolder, located in the same **_MSP Builder** folder created by the RMM Suite installation.

- 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.



```
_MSP Builder
  Core Automation
  EMM Procedures
    Maintenance
      WIN-MB Maint (MV)
    Maintenance Utilities
      WIN-MB Maint - Health Check
      WIN-MB Maint - Set DoNotClearRecycleBin Flag
      WIN-MB Maint - Set GuiSilentStart Flag
      WIN-MB Maint - Set SkipMaintReboot Flag
      WIN-MB Maint - Upload All Maint Logs as Zip
      WIN-MB Maint - User GUI - Deploy to Startup
      WIN-MB Maint - User GUI - Remove from Startup
      WIN-MB Maint - Workstation - Immediate (MV)
      WIN-Send User Message (prompt)
      WIN-Send User Message + Response (prompt)
    Smart Monitors
      WIN-MB Internet Gateway Failover Detection
      WIN-MB Smart Monitors
    System Utilities
      WIN-MB Deploy EMM Config Files
```

Maintenance

The Daily Maintenance procedure is located here. The procedure initiates maintenance on both servers and workstations but performs some additional steps on workstation platforms to recycle the user interface. By default, the maintenance is scheduled between 6 and 8am for servers and between 10am and 4pm for workstations.

Maintenance Utilities

These procedures are used to maintain or alter the configuration of the Daily Maintenance tools. There is also a special procedure here for workstations that will invoke all maintenance tasks immediately. This will cause the missing maintenance tasks to be run without any time of day restrictions. This procedure is also run via ITP in response to end-user

Smart Monitors

The **WIN-MB Smart Monitors** procedure performs the deployment and execution of all Smart Monitors and is deployed equally to all platforms. The Smart Monitors themselves will limit their action to server or workstation platforms as appropriate. Also located here is the Smart 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.

Monitors

Monitor sets related to the alerts triggered by the EMM tools are provided. The monitor sets are applied automatically through "Auto-Pilot" System Policies.

Overview

All Daily Maintenance and Smart Monitor alerting is done via the Event Log. Five event log monitors are provided for Maintenance tools, and four monitors cover the Smart Monitors.

Maintenance Monitors

MB-MNT-A.EVT.S.P3.Alm
This monitor tracks alerts from the various maintenance tools deployed to Servers. These generate priority 3 type alerts, including invalid task definitions, excessive uptime, disk defrag failures, and disk-check alerts. None of these are capable of auto-remediation.

MB-MNT-A.EVT.S.P4.Req
This is a special server monitor that interfaces with the User GUI. When the GUI is deployed and run on server platforms, an additional button appears that allows the technician to suspend monitoring/alerting for 2 hours. This alert can also be triggered via the command line tool **Suspend.bat**.

MB-MNT-A.EVT.W.P3.Alm
This is the core maintenance monitor for workstation events. In addition to the base set of monitors defined in the server monitor set, it adds alerts related to failures of the User GUI, local backups, and creation of System Restore Points.

MB-MNT-A.EVT.W.P3.Chk
This is an optional monitor set that can be configured to report when nightly maintenance has failed to run for a specific number of cycles. This is not deployed by default as users are notified automatically when maintenance isn't completed, and they are given the opportunity to invoke maintenance themselves.

MB-MNT-A.EVT.W.P4.Req
This is a special workstation monitor that is triggered by the User GUI when the user requests immediate execution of missed maintenance tasks. This monitor will also be used to respond to customized events from the User GUI.

Smart Monitors

MB-SM-A.EVT.S.P3.Act
This is an actionable event that triggers a remediation procedure when the Time Service is incorrectly configured. The remediation procedure applies the correct settings and updates the service.

MB-SM-A.EVT.S.P3.Alm
This monitor set creates alerts when a server reboots during business hours; boots and remains in Safe Mode; fails to calculate uptime; disk capacity (immediate and trending); and time sync issues that cannot be auto-remediated.

MB-SM-A.EVT.W.P3.Alm
A monitor set for workstations that reports on Antivirus issues, disk capacity (immediate and trending), and time sync issues that can't be auto-remediated.

MB-ICC-A.EVT.S.P3.Alm
A monitor set that works with the Internet Connection Check Smart Monitor. It alerts on failover, failback, and configuration issues. It can also trigger an error when deployed to an unsupported platform, since this monitor is deployed to a specific target within a customer environment as a result of manually running the procedure to install the monitor.

EMM Procedures

EMM uses Kaseya Procedures to deploy and execute the actual Maintenance and Smart Monitor applications. The applications then run without dependence on Kaseya, reporting status where necessary through alerts.

- **WIN-Agent Daily Tasks - <type>**
  These procedures define the agent type and then initiate the **WIN-Agent Daily Tasks** procedure. The WIN**-Agent Daily Tasks** procedure does the actual work.
  - Deploys the MSP_Identity.ini file, which controls license authorization and certain settings used by the Daily Maintenance and Smart Monitor tools, as well as the End-User Interface.
  - Sets global data values and ensures that all necessary tools and applications are downloaded and up-to-date.
  - Deploys the Daily Maintenance and Smart Monitor
  - Initiates Smart Monitors via the **WIN-MB Smart Monitors** procedure.
  - Initiates Maintenance by the **WIN-MB Maint** procedure.
  - The daily Audit procedure is run on all systems, including those in audit and unmanaged groups, unless the DAILY Policy Control value is set.
- **WIN-MB Deploy EMM Config Files**
  Deploy the configuration files for Daily Maintenance and Smart Monitors.
  - Removes all prior configuration files.
  - Deploys the standard configuration files.
  - Deploys the customer-specific configuration files, if present.
  - Deploys the group-specific configuration files, if present.
  - Deploys the machine-specific configuration files, if present.

  Note that this procedure performs “blind copies” of the configuration files. The agent may report several file copy failures. This is expected when custom configuration files are not present, and this error can be safely ignored

- **WIN-MB Smart Monitors**
  Initiates the Smart Monitors. This procedure skips systems if the EMS value is false, the DisableEMM managed variable is true, the SMON Policy Control value is set, or where the agent is a member of an Audit or Unmanaged group. The Disk Capacity checks are initiated regardless of the EMS and DisableEMM values as these are part of the Core Automation components.
- **WIN-MB Maint**
  Initiates the Daily Maintenance sequence engine. This procedure skips systems if the EMS value is false, the DisableEMM managed variable is true, the MAINT Policy Control value is set, or where the agent is a member of an Audit or Unmanaged group..
- **WIN-MB Maint – Workstation – Immediate (MV)**
  This procedure causes all missed day and night tasks to be run immediately. This procedure can be initiated by VSA admins to manually invoke the missed maintenance. It is also executed by ITP in response to the end user clicking the “Run Maintenance Now” button. This automation is dependent on the Intelligent Ticket Processing service running and being properly configured to map the alert to this procedure.

- **WIN-MB Maint - User GUI – [Deploy to | Remove from] Startup**
  This procedure will either deploy the End User GUI to the active user's startup folder or remove it from the startup folder. This controls whether the GUI is initiated at system logon.
- **WIN-MB User GUI – Start**
  This procedure evaluates the system and org settings to determine if (and how) the User Interface should be started. *This procedure cannot be run directly as it depends on settings defined in other procedures!*
- **WIN-MB Maint - Health Check**
  A procedure that will verify that the required tools are present and mark the agent's Health Status field with Pass/Fail. Unsupported platforms are set to "N/A". Note that these tasks are normally performed automatically as part of the Daily Audit.
- **WIN-MB Internet Gateway Failover Detection**
  This is used to deploy a special Smart Monitor that monitors an environment which has redundant Internet connections. An alert is generated when the connection switches between primary and secondary. Running this procedure installs and configures the monitor.
- **WIN-Send User Message [+ Response] (Prompt)**
  Provided for integration in custom procedures, these procedures leverage the End User Interface to display a message using the MSP's customized interface. The standard procedure displays the message with an "OK" button, while the "+ Response" version displays both "Yes" and "No" buttons. The
- **WIN-MB Maint – Set <option>**
  These procedures set specific options on the endpoint, allowing per-agent control over these settings.
- **WIN-MB Maint – Upload All Maint Logs as Zip**
  This will add all of the log files from the Logs folder into a zip file and upload it to the agent's GetFiles location. This procedure should always be run when requesting support for MSP Builder applications. The zip file can be attached to an email request or uploaded to the online ticket submission form.

Smart Monitors

MSP Builder's Smart Monitor technology is designed to improve monitoring while reducing alerts that reach the MSP's help desk. They accomplish this by deploying local intelligence specially designed to monitor and remediate specific conditions. Some Smart Monitors work only on Server or Workstation platforms, while others are universal. Currently, Smart Monitors only support the Windows operating system platform.

All Smart Monitors are designed to fit the widest array of conditions using reasonable and sane default values. Configuration files can be created and deployed, either globally to accommodate MSP-specific defaults, or by specific customer or machine to accommodate unusual or "one-off" conditions. The design is such that customization is minimized, reducing and often eliminating manual configuration. When a configuration file is employed, it will always be located in the Kaseya Work directory.

Smart Monitors are deployed and initiated daily by the Daily Tasks procedure. Some Smart Monitors run just once a day, others run on a schedule throughout the day. All* smart monitors have a 24-hour maximum cycle, so that they will exit prior to the next day's invocation.

**The Server Boot Monitor employs an "at startup" scheduled task. This task is removed and recreated daily but will remain present if the Smart Monitor is removed. The scheduled task should be removed manually to prevent further reboot alerts. This can be done by running the RMSSBM.BMS script manually and specifying the --r parameter.*

While all Smart Monitors run independently of the RMM platform, they can generate alert events to notify the RMM of specific conditions. When Smart Monitors are deployed, the appropriate monitor sets are also linked to the agent so that the events will be monitored. Likewise, monitor sets are removed when Policy Control is used to disable the Smart Monitors.

Smart Monitors are monitored by the **MB-SM-A.EVT.S.P3.Alm** or **MB-SM-A.EVT.W.P3.Alm** monitor sets (server and workstation, respectively). The monitor sets will include only alerts that are platform appropriate, since some Smart Monitors are Server or Workstation specific.

Configuration Files

Most Smart Monitors use reasonable default values, limiting the need for configuration files to be deployed. The actual configuration process is very similar to that of the EMM Suite Maintenance utility, so the configuration process will be combined in a later chapter. The primary difference is the location of the configuration files – MSPB_EMM\SMon_Cfg instead of the MSPB_EMM\Maint_Cfg. Global configuration files are placed in the "_Common" subfolder, while customer, site, and agent-specific folders are created directly in this Configuration folder.

Alerts

All Smart Monitors generate alerts via the Event Log. Most Smart Monitors will write an Informational event to the log when they execute, and no issues/errors are detected. These informational events are not monitored in the default configuration and are provided simply to verify operation. A monitor set can be developed and deployed to report on *missing* informational events to alert when these Smart Monitors are not running as expected. This is not recommended for general operation but may be useful in diagnosing delivery and execution issues with particular agents or customers. A special type of monitor is provided to track the successful self-remediation of the Smart Monitors. These alerts are collected by ITP to provide the MSP with statistical information and will not result in PSA tickets.

Note that all "400-series" alerts are related to licensing – contact support if these errors are widespread or persist on agents for more than 1 day.

RMSSSC – System Security Check

Determines the state of the AV & security products, alerting if missing, outdated or not running.

Summary

This is a Workstation-only utility that uses Microsoft's Security API to determine the AV product(s), status, and related information. The API is not available on server platforms, so this monitor will exit silently when a server operating system is detected.

Alerting

The following alerts are generated by this Smart Monitor.

Class	Event	Source	Message
INFO	110	RMM-SMARTMON	RMSSSC: Security Checks Performed.
INFO	110	RMM-SMARTMON	RMSSSC: STATUS: <message>
ERROR	111	RMM-SMARTMON	RMSSSC: Antivirus - Product not detected!
ERROR	112	RMM-SMARTMON	RMSSSC: Antivirus - Outdated - <AV_NAME>
ERROR	113	RMM-SMARTMON	RMSSSC: Antivirus - Not running - <AV_NAME>
ERROR	114	RMM-SMARTMON	RMSSSC: Antivirus - Multiple products are running!
ERROR	115	RMM-SMARTMON	RMSSSC: Antivirus - Protection is suspended - <AV_NAME>
ERROR	116	RMM-SMARTMON	RMSSSC: Antivirus - Preferred Product not installed!

Transient Suppression

Varies – most conditions alert immediately and then suppress additional alerts, limiting them to once every 3 days, while the Outdated alert (112) will not alert for 48 hours while self-remediation attempts to resolve the issue, then will alert every 3 days thereafter until resolved.

Self-Remediation

If the detected AV definitions are not current, the commands defined in the configuration file for the identified product are run to initiate a manual definition update. In many cases, the next check will find that the definitions are current and no alert will be generated. When the update fails, an alert will be generated when the transient suppression period expires. This often indicates that the agent is incapable of accessing the definition update URL.

Arguments

--s run silently, suppress all screen output. This monitor normally displays status on the console window.

--d enables debug logging – additional status messages are displayed and logged. The MSP can run this utility locally in Debug mode to display the status data used to determine various conditions.

--a display status (audit) – used with other automation to report whether the condition(s) that caused an alert still exist.

--p:name specify the preferred product. Run the RMSSSC.BMS script interactively to see the list of product names, or check the latest log file.

Configuration File & Parameters

The configuration file for this Smart Monitor named "**RMSSSC.ini**" and is *required* for operation. It can be used to create overrides like other Smart Monitors, but there are two required parts of data that are defined in the configuration file. The first is the Preferred Product (which is usually MSP-specific) and then one or more Remediation definitions.

COMMON Section

This section controls the operation of the Smart Monitor.

Preferred The name of the preferred antivirus product. When not defined, error 116 is never triggered by the monitor.
When the AV product experiences a name change, you can list the old and new names, separated by a comma. This should only be done when a single AV product reports multiple names to Microsoft Security Center.

SkipAll If true, all alerts are suppressed.

SkipEventID_# If true, the alert ID specified is suppressed. Multiple individual alerts can be suppressed by specifying individual SkipEventID values. See the list of alert code numbers under the Alerting section above.

Product_AV

This section is named after the product and must include the "_AV" suffix. A section can exist for each of multiple AV products, and define the steps to take to identify and then run the process to manually update the definitions. It supports the following configuration values:

Type Either REG or CMD, determining whether to obtain the command from a PATH## or KEY## parameter.

Path## The disk path where the command is located. "##" is either 32 or 64 and represents the value for 32 or 64-bit platforms.

Reg## The registry path where the command path is defined. "##" is either 32 or 64 and represents the value for 32 or 64-bit platforms.

RVal The registry value to read to obtain the command folder path, or use "#ENUM#" to enumerate the values in the key.

Cmd The command to run after appending it to the path extracted from the Path or Key values above.

Arg The argument(s), if any, to pass to the command to initiate a definition update.

Response A key word or phrase to check for in the response to determine if the request was successful.

RMSSBM – Server Boot Monitor

Alerts when a server boots during business hours or into DSRM or Safe Mode.

Summary

This Smart Monitor will alert when a server boots into Safe Mode and then remains in Safe Mode for more than 15 minutes. This alert is triggered after any reboot. It also alerts when a server is rebooted during normal business operating hours – 8am to 9pm local time by default.

Alerting

The following alerts are generated by this Smart Monitor.

Class	Event	Source	Message
ERROR	121	RMM-SMARTMON	RMSSBM: The system has booted into Safe Mode.
ERROR	122	RMM-SMARTMON	RMSSBM: The system has booted into Directory Services Restore Mode
ERROR	123	RMM-SMARTMON	RMSSBM: The system has Safe Mode Boot enabled
ERROR	124	RMM-SMARTMON	RMSSBM: One or more monitored services have failed to start
ERROR	126	RMM-SMARTMON	RMSSBM: The system has booted during business hours.
ERROR	129	RMM-SMARTMON	RMSSBM: Unable to calculate uptime.

Transient Suppression

None – all alerts are triggered upon detection. The Safe Mode alerts are suppressed for a defined period (default – 15 minutes) after system boot to allow time for an engineer to perform a controlled boot into Safe Mode, perform a remediation task, and then reboot normally. The intent is to alert on unintentional or uncontrolled booting into Safe Mode in response to a boot-detected issue. Event 123 will alert that the next reboot of the server will cause it to start in Safe Mode. This is a predictive alert and not indicative of the current system state.

Self-Remediation

None.

Arguments

--d enables debug logging – additional status messages are displayed and logged.

--c Enables the checks. Without this argument, it creates the “at startup” scheduled task.

--r Removes the Startup Task and exits.

Configuration File & Parameters

The configuration file “**RMSSBM.INI**” is used to alter the Safe Mode detection timer, the business operational hours, and to include Weekends in the business hour time period.

COMMON Section

SMDetectTimer The time delay (in minutes) after booting into safe mode before an alert is generated. The default value is 15 minutes, which is also the minimum allowed value. When changing this value, it is not recommended to use values above 60 to ensure effective alerting.

BusinessHours A comma-delimited pair of HH:MM values that define the start and end of what should be considered “business hours”. The default range is 8am to 9pm (08:00,21:00), Monday through Friday.

Weekend A Boolean (T/F) value that, when True, includes Saturday and Sunday in the business hours reboot alerting process.

SERVICES Section

ServiceName A Boolean (T/F) value that, when True, generates an error if the named service is not running after the startup delay timer has expired.

RMSDCC – Smart-Logic Disk Capacity Check

Performs hourly disk capacity checks and once-daily trending analysis.
This Smart Monitor is part of the Core Automation components and is always deployed even when EMM components are not used.

Summary

This smart monitor intelligently checks each detected volume on a system for available capacity. The check is based on one of 9 “container” sizes, which allows different threshold factors to be used to determine the optimal alerting threshold. With this design, as the disk size increases, the threshold decreases proportionally. Volumes below a certain size or having specific volume names are excluded from monitoring. Workstations and Servers both have a unique set of container sizes, generating the most appropriate thresholds for each platform type.

The monitor supports both volumes with assigned drive letters and mounted volumes, removing the limitation of monitoring by drive letter.

When the monitor starts each day, it updates the prior 30 days of utilization data for every volume. The trending of space utilization is then projected 30 days into the future to determine if the threshold will be crossed at the current usage rate. If so, a warning event will be generated.

Alerting

The following alerts are generated by this Smart Monitor.

Class	Event	Source	Message
INFO	160	RMM-SMARTMON	RMSDCC: Disk Capacity Check passed.
INFO	160	RMM-SMARTMON	RMSDCC: STATUS: <message>
ERROR	161	RMM-SMARTMON	RMSDCC: Disk Capacity Trend Alarm for <D:>. Anticipate full utilization in 28 days at current consumption rate.
WARNING	162	RMM-SMARTMON	RMSDCC: Disk Capacity Warning. Free space is near the minimum threshold on one or more drives.
ERROR	162	RMM-SMARTMON	RMSDCC: Disk Capacity Alarm. Free space is below minimum threshold on one or more drives.
ERROR	163	RMM-SMARTMON	RMSDCC: Disk Capacity CRITICAL Alarm. Free space is >85% below minimum threshold on one or more drives.
WARNING	165	RMM-SMARTMON	RMSDCC: Disk Monitors suspended on drives: <list> (alert only with --q argument)
ERROR	169	RMM-SMARTMON	RMSDCC: Invalid custom parameter specified

Transient Suppression

Yes – Alerts are suppressed for 48 hours unless the utilization crosses the threshold by 65%, which will generate an immediate alert. Warnings for projected utilization are triggered once upon detection. If the combination of Transient Suppression and Self-Remediation resolves the alert condition, the successful remediation is logged with Event 160 and the event details in the <message> body.

A disk volume must be “seen” for more than 6 consecutive check cycles before it is considered “permanent” and monitoring will be permitted. This will prevent temporary volumes mounted during virtualization-aware backups from reporting a low-space condition. The “seen” count can be adjusted via the configuration file.

Self-Remediation

Yes – upon detection of disk capacity below the calculated threshold, the EMM Maintenance tool “RMMSCU.BMS” script will be invoked to initiate a cleanup of all temporary file locations, using an argument that reduces the file age to 1 day. Note that any additional folders defined by a local configuration file will also be examined for file cleanup.

Arguments

Many of the available arguments are intended for an engineer to use when running interactively, either to generate a report, diagnosing a configuration, or modifying the operation. The procedure that executes this Smart Monitor can also be modified to include any appropriate argument.

--a	Audit - Report 0 if no issues detected, 1 if space alerts exist.
--d	Enable Debugging messages.
--w	Enable Warning messages to be logged (default is Alert only).
--r	Report on allowed drives.
--q	Generate alert if drive monitors are suspended.
--st	Create the Scheduled Task.
--su d: #	Suppress reporting for the specified volume. "#" represents the number of days to supppress reports.
--t	Turn off "Tiny Drive" exclusions - report on drives of all sizes.
--c:<d>	Reset (clear) disk trending data for the specified volume name.

Configuration File & Parameters

The configuration file "**RMSDCC.INI**" is used to provide general and volume-specific overrides for the container sizes, threshold factors, or both. As the common and volume-specific parameters are the same, they will be covered only once.

COMMON Section

The common section contains parameters that control the configuration of the Smart Monitor, including settings that apply to all volumes.

ExcludeTinyDrives A Boolean value that can enable checking of "tiny" drives. By default, any disk volume with a size below 18 GB is ignored. This prevents alerting on what are typically "recovery" volumes. Set this value to "Y" to enable checking all volumes regardless of their size.

TinyDriveSize Changes the "Tiny Drive" default value of 18 GB to a custom size, in GB.

NoWarn Turn off trending warnings.

LabelExclusionList Adds a comma-delimited list of volume labels to the default list of "recovery" and "HP Tools". Any volume label containing any of these terms will be excluded from capacity checks.

TDSThreshold Defines an override for the Temporary Drive Sighting Threshold. The default is 6, so a volume will not generate an alert until the 7th hourly scan. This prevents temporary volumes mounted during backup operations from being alerted upon. This count also prevents the first 6 (or otherwise defined) monitoring cycles from generating an alert for any disk volume.

DiskFactor The comma-delimited list of values that sets the "container" size used to apply a threshold calculation factor. There are 9 values in this list, and all must be specified. The default values are 99, 299, 499, 999, 2499, 3999, 5999, 7999, and 99999 for servers and 49,99,149,199,249,299,499,749,9999 for workstations. The actual volume size is compared to determine which size range (container) it belongs to. For example, a workstation volume of 160GB is greater than 149 but less than 199, so it is in "container 3". The matching SizeFactor value is then used to compute a free-space threshold against the actual volume size. This design allows container sizes to be adjusted for typical workstation and server systems.

SizeFactor The value that is used in the internal calculation to determine the free space threshold. It works in conjunction with the DiskFactor value. The default values are 9, 11, 14, 17, 20, 24, 26, 30, and 32. Using the prior example of a 160 GB volume, it would fit into Container 2. The corresponding size factor value is “11”.

The SizeFactor value is the most often adjusted parameter and is usually customized when a system has a volume close to the threshold and no plan exists to increase the available space. The DiskCapCalc.xlsx spreadsheet, provided with the EMM Suite and available for download from the mspbuilder.com website, will assist in determining the appropriate override value to use. The spreadsheet itself has detailed instructions for use.

Note that both the DiskFactor and SizeFactor parameters require a list of 9 values. An incorrectly formatted parameter will generate an Event 169 alert, and the default values will be applied.

DisableWS A Boolean value that, when True, disables processing of this Smart Monitor on workstation class systems.

VOLID Section

When a server has multiple volumes of a similar size, it may not be appropriate to modify the common parameters for DiskFactor and SizeFactor. In this case, a volume-specific section can be created and the SizeFactor value defined. In this case, it will apply only to the named volume. Disks are specified as “D:/” (including the slash), and mounted volumes are defined by their mount path (C:/mount/data). Note the use of *forward slash* in this parameter!

When defining the VOLID section, only the DiskFactor and/or SizeFactor parameters can be provided. Any other values will be ignored.

Volume ID Exclusions

Volumes can be excluded by defining an “EXCLUDE” section in the configuration file. This section will contain VolumeID values with a Boolean value that will cause the named volume to be excluded from processing when the value is “True”. Any Boolean value that evaluates to True can be used, including “T”, “Y”, or “1”.

The Volume ID of any valid volume is reported in the logs, making the name easy to identify. This allows the exclusion of drives, mounted volumes, and unmounted volumes that are active. All three types are shown in the example below:

```
[EXCLUDE]
C:/=Y
C:/Mount/LocalData/=Y
Volume{57eae949-0000-0000-0000-606a25000000}=Y
```

RMSUSC – User & Group Security Check

Generates warning alarms when changes to user accounts and groups are detected, both local and in Active Directory.

Summary

This monitor will keep track of local and (on the PDCe) Active Directory groups and generate an alert when changes are detected. Changes include adding a group, removing a group, adding an object to a group, or removing an object from a group. The group, action, and all changes are reported in the alert. Only ONE alert is generated for any number of detected changes per pass of the Smart Monitor.

Alerting

The following alerts are generated by this Smart Monitor.

Class	Event	Source	Message
WARN	171	RMM-SMARTMON	RMSUSC: <messages>
ERROR	172	RMM-SMARTMON	RMSUSC: <messages>
WARN	173	RMM-SMARTMON	RMSUSC: <messages>

Changes to groups or user accounts will generate a 171 Warning event. Accounts added to the local Administrators group or any of the standard admin groups in AD (domain admins, schema admins, or enterprise admins) will generate a 172 Error event. ONE event is generated regardless of the number of changes detected; however all changes are logged in the event message body. A 173 Warning event is generated when an account lockout condition is first detected.

The AD users/groups are checked only on the PDCe host.

Messages

The following messages are written as part of the alert. One alert may have several messages.

GROUP ADDED: <name> - USERS: <list>
GROUP REMOVED: <name>
USER ADDED: <user> IN GROUP: <name>
USER REMOVED: <user> FROM GROUP: <name>

Transient Suppression

There is no suppression of events in this Smart Monitor. All changes are reported immediately.

Self Remediation

No self-remediation actions are performed.

Arguments

No arguments are used by the Smart Monitor.

Configuration File & Parameters

The configuration file for this Smart Monitor is optional and can be used to define exclusions for trusted accounts. A built-in exclusion is the LAUser account. It can also allow select customizations to operation.

COMMON Section

Exclude A comma-delimited list of account names to exclude from the detection process.

DelayInit Allows immediate execution after first-time run if set to “N”. Normally, the Smart Monitor will not run for the first 48-hours after onboarding or first execution.

SuppressNonAdmin Suppresses Event 171 alarms for changes to non-admin accounts.
SuppressNonAdmin=Y suppresses all 171 events

SuppressNonAdmin_W Suppresses the alarm only on workstations.

SuppressNonAdmin_S Suppresses the alarm only on servers. Suppressing alarms on servers is not recommended to maintain effective security awareness.

RMSNTP – Network Time Check

Verifies the local system time against the local domain, and against public NTP if running on a PDCe.

Summary

This monitor will check the Windows Time Sync service and verify that it is configured according to Best Practices and within synchronization limits. When remediation is enabled the Smart Monitor will reset the time service parameters and/or initiate a resync.

There are two configuration settings considered "correct" by Microsoft. The PDCe should be configured to use NTP for synchronization, and all other member devices should use NT5DS. When the Smart Monitor detects a PDCe configured for NT5DS, it reconfigures it to use NTP, and assigns three separate time hosts. The default time hosts are 0, 1, and 2.us.pool.ntp.org. In regions outside of the US, or organizations with a specific internal time hierarchy, the Smart Monitor deployment procedure should be updated so that the --t:*list* argument provides the desired time server hosts. For non-PDCe systems running NTP, the configuration is reset to use NT5DS.

Alerting

The following alerts are generated by this Smart Monitor.

Class	Event	Source	Message
INFO	180	RMM-SMARTMON	RMSNTP: All time values are within spec.
INFO	180	RMM-SMARTMON	RMSNTP: STATUS: <message>
WARNING	181	RMM-SMARTMON	RMSNTP: Warning - Domain Time: <message>
ERROR	181	RMM-SMARTMON	RMSNTP: Alert - Domain Time: <message>
ERROR	182	RMM-SMARTMON	RMSNTP: Alert - NTP Time: Difference between Domain and NTP is greater than 2 minutes after resync!
ERROR	184	RMM-SMARTMON	RMSNTP: Alert - Multiple NTP Time Resync actions within 7 days!
WARNING	185	RMM-SMARTMON	RMSNTP: Warning - PDCe is not using NTP
WARNING	186	RMM-SMARTMON	RMSNTP: Warning - Member server is not using NT5DS
WARNING	187	RMM-SMARTMON	RMSNTP: Alert - W32Time service is missing.

Transient Suppression

Alerts are suppressed only if remediation is enabled and successful. This includes correcting the time sync configuration or performing a resync with the upstream server(s). If the combination of Transient Suppression and Self-Remediation resolves the alert condition, the successful remediation is logged with Event 180 and the event details in the <message> body.

Self-Remediation

Yes – If the time configuration is not correct, it is updated (NTP on PDCe only, NT5DS on all other devices). If the time is out of sync with the domain, a resync is performed.

Arguments

--V	Verbose – write extra information to log/console.
--tt	Abort if wrong time sync type (ie: do not alert if using NTP on a member server).
--s	Suppress alerts, just log results.
--a	Audit mode – report if a fault is present.
--t:l	List of alternate NTP hosts to use for PDCe configuration. The default is 0-2.us.pool.ntp.org (3 separate host definitions).

Configuration File & Parameters

The configuration file for this Smart Monitor is optional and is used to override default settings. The file name is "**RMSNTP.INI**" and contains 4 parameters in the COMMON section.

COMMON Section

Debug Operate in DEBUG mode – provide verbose console output and logging. Console output is usually suppressed or minimized.

NtpHosts List of alternate NTP hosts, same as --t: command-line argument. Unless using a GPS or Radio Clock, at least three separate time sources should be specified.

IgnoreW32Time Bool - ignore service validation if true (also - IgnoreW32Time RegKey). When this is set, only the comparison of local and upstream server time is performed. This might be used when a custom time service is installed.

DontFixSvc Bool - don't fix the time service. Alerts are generated for non-compliant configurations, but no changes to the service configuration are made.

ResyncDelay Number of seconds to wait after a resync before re-checking - 90s default. If the time is still not in-spec with the upstream server after this delay, an alert event 182 is generated.

RMSICC – Internet Connection Failover Check

Reports when an Internet connection transitions from Primary to Backup or vice-versa.

Summary

This is a special form of Smart Monitor that is not deployed globally. A procedure is used to deploy this to a specific host in a client environment. The initial deployment should be performed when the customer is using their primary Internet connection. Once deployed, the Smart Monitor is permanently deployed and runs on a cycle to report transitions between primary and backup connections. These alerts are monitored as warnings since the failure doesn't represent an outage. Many customers have services bound to a specific interface (VPN tunnels, for example) and need to be notified of transitions. The Smart Monitor will generate alerts for invalid configurations or when it is deployed to an unsupported (outdated or workstation) platform.

Alerting

The following alerts are generated by this Smart Monitor.

Class	Event	Source	Message
WARNING	131	RMM-SMARTMON	RMSICC: On Primary Connection
WARNING	132	RMM-SMARTMON	RMSICC: On Backup Connection
ERROR	135	RMM-SMARTMON	RMSICC: Primary IP not set - NOT CONFIGURED!
ERROR	136	RMM-SMARTMON	RMSICC: Unsupported Platform!

Transient Suppression

None.

Self-Remediation

None.

Arguments

--d	Enable Debugging messages.
--SET	Set/update the primary IP by detection of current external IP address with confirmation.
--SETA	Force an update of the primary IP by detection of current external IP address.

Configuration File & Parameters

None. The active configuration is maintained in the registry at HKLM\SOFTWARE\RMM. Aside from setting the Primary Interface IP Address, there are no configurable parameters.

Event IDs Assigned to Smart Monitors

Each Smart Monitor is assigned a range of Event IDs. The Event Log messages are defined below, along with the appropriate response to detection of the event. Event IDs ending with zero generally denote a status message that indicates that the task has run successfully. These are informational messages and are not alerted on.

All event log source values are “RMM-SMARTMON”.

Event Type Codes:

0	Success
1	Error
2	Warning
4	Information

Component	Source	Type(s)	Event	Message
RMSSSC	Endpoint Security Check	4	110	Security Checks Performed or STATUS: <message>
RMSSSC	Endpoint Security Check	1	111	Product not detected!
RMSSSC	Endpoint Security Check	1	112	Outdated definition
RMSSSC	Endpoint Security Check	1	113	Not Running
RMSSSC	Endpoint Security Check	1	114	Multiple products are running
RMSSSC	Endpoint Security Check	1	115	Protection is suspended
RMSSSC	Endpoint Security Check	1	116	Preferred product not installed
RMSSBM	Server Boot Monitor	1	121	The system has booted into Safe Mode
RMSSBM	Server Boot Monitor	1	122	The system has booted into Directory Services Restore Mode
RMSSBM	Server Boot Monitor	1	123	The system has Safe Mode Boot enabled
RMSSBM	Server Boot Monitor	1	125	One or more monitored service have failed to start
RMSSBM	Server Boot Monitor	1	126	The system has booted during business hours
RMSSBM	Server Boot Monitor	1	129	Unable to calculate uptime
RMSICC	Internet Failover Monitor	2	131	On Primary Connection
RMSICC	Internet Failover Monitor	2	132	On Backup Connection
RMSICC	Internet Failover Monitor	1	135	Primary IP not set – NOT CONFIGURED
RMSICC	Internet Failover Monitor	1	136	Unsupported Platform!
RMSDCC	Disk Capacity Check	4	160	Disk Capacity Check passed or STATUS: <message>
RMSDCC	Disk Capacity Check	1	161	Disk Capacity Trend Alarm for <d:>
RMSDCC	Disk Capacity Check	2	162	Disk Capacity Warning
RMSDCC	Disk Capacity Check	1	162	Disk Capacity Alarm
RMSDCC	Disk Capacity Check	1	163	Disk Capacity CRITICAL Alarm
RMSDCC	Disk Capacity Check	2	165	Disk monitors suspended on drives:
RMSDCC	Disk Capacity Check	1	169	Invalid custom parameter specified
RMSUSC	User Security Check	2	171	RMSSBM: <message> reporting changes to users and groups
RMSUSC	User Security Check	1	172	RMSSBM: <message> reporting changes to admin users and groups
RMSUSC	User Security Check	2	173	RMSSBM: <message> reporting account lockouts
RMSNTP	Network Time Check	4	180	All values within spec or STATUS: <message>
RMSNTP	Network Time Check	2	181	Warning – Domain Time:
RMSNTP	Network Time Check	1	181	Alert – Domain Time:
RMSNTP	Network Time Check	1	182	Alert – NTP Time: Difference > 2 minutes after resync
RMSNTP	Network Time Check	1	183	Alert – NTPDATE.EXE is not present on client! Fatal error!
RMSNTP	Network Time Check	1	184	Alert – Multiple NTP Time Resync actions within 7 days
RMSNTP	Network Time Check	2	185	Warning – PDCe is not using NTP
RMSNTP	Network Time Check	2	186	Warning – Member server is not using NT5DS
RMSNTP	Network Time Check	2	187	Alert – W32Time service is missing

RMM Maintenance

The EMM Suite provides a mechanism for performing daily maintenance tasks on agent endpoints. The maintenance tasks are performed without the need for continuous communication with the VSA platform. This ensures that maintenance tasks are completed even when the agents are off the network.

There are three primary components used by the Maintenance process.

Sequencing Engine

The key component of Maintenance is a “Sequencing Engine”. This application is deployed by the VSA along with a configuration file, and then invoked. The Sequencing Engine then takes over, validating the configuration data, performing maintenance tasks, scheduling maintenance tasks for after-hours, and reporting any issues with the maintenance process. It supports several built-in capabilities for rebooting, displaying messages to the user, checking uptime, and creating scheduled tasks.

The Sequencing Engine can execute any command as part of its maintenance regimen, from batch scripts to executables, allowing for a high degree of flexibility and customization. It can execute commands on schedules ranging from Daily to specific days of the week or month. The exact time of daily execution cannot be controlled using the default methods, but the Maintenance procedure can be precisely scheduled through the VSA where required.

Maintenance runs daily on two distinct cycles. The first invocation is when the VSA deploys the daily maintenance. This cycle usually performs non-invasive tasks such as local file backups, folder cleanups, and various checks, plus scheduling the evening cycle. This cycle often completes in just 30-40 seconds. The second cycle is scheduled to run later in the day at a fixed time. Maintenance tasks that might be noticed by the end user, such as disk checks or defrag operations. Most evening tasks also run fairly quickly and choosing a time just prior to the end of the work day is a good idea to ensure that the largest number of workstations are available to complete these tasks.

Maintenance Tools

The EMM Suite provides several standard maintenance tools, as described in this chapter. New tools are being developed constantly based on customer feedback and direct experience with our MSP clients. The maintenance process is not limited to these tools, as any script, batch file, or executable can be included in the daily maintenance configuration.

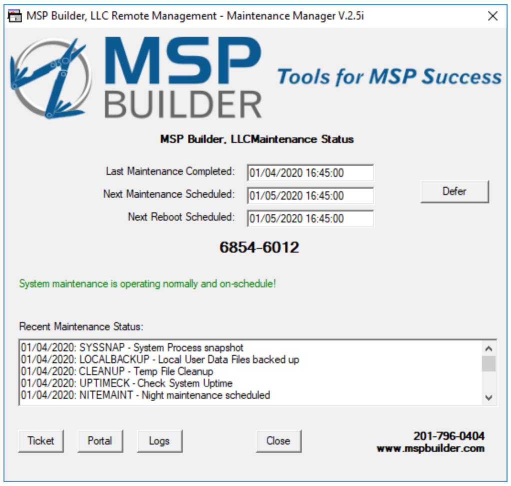
The standard tool kit consists of the following utilities:

- Temp File Cleanup Scans all system and user TEMP locations for outdated files and removes them after 5 days.
- Volume Defrag Utility Invokes the standard Windows Defrag (or a custom tool provided by the MSP) to defragment the System and Data drives on unique schedules. This utility is SSD-aware and prevents running defrag operations on SSD drives.
- Local File Backup Creates a local backup of key user files such as Templates, Shortcuts, Favorites, and desktop objects.
- Disk Integrity Check Runs both a CHKDSK command to verify the disk integrity and a SMART test to extract predictive failure data from supported disks.

There are also several “utility” applications included in the EMM Suite that can be used by the Sequence Engine. Many of these set or report configuration parameters and are primarily designed for ad-hoc use but may be appropriate for regular maintenance use.

User Interface

The User Interface is a small, lightweight GUI application that keeps the user informed of when maintenance tasks will run, when the next reboot is scheduled, and reports the last 50 maintenance events. It also provides links to the MSP's web site, portal, and provides contact information. It also provides the user with the ability to defer maintenance for 24 hours or to run maintenance tasks that were missed. It also provides a mechanism for communicating between the VSA and the end user with dialog boxes sporting the MSP's logo instead of a simple grey box. Messages can be scheduled through the Sequence Engine or delivered via Kaseya procedures.



MSP Builder, LLC Remote Management - Maintenance Manager V.2.5i

MSP Builder, LLCMaintenance Status

Last Maintenance Completed: 01/04/2020 16:45:00
Next Maintenance Scheduled: 01/05/2020 16:45:00
Next Reboot Scheduled: 01/05/2020 16:45:00

6854-6012

System maintenance is operating normally and on-schedule!

Recent Maintenance Status:

```
01/04/2020: SYSSNAP - System Process snapshot
01/04/2020: LOCALBACKUP - Local User Data Files backed up
01/04/2020: CLEANUP - Temp File Cleanup
01/04/2020: UPTIMECK - Check System Uptime
01/04/2020: NITEMAINT - Night maintenance scheduled
```

201-796-0404
www.mspbuilder.com

The interface is fully customizable with the MSP's masthead, logo, contact information, and URLs.

Controlling the Interface Display

The interface will display briefly (about 6 seconds) when the user logs in, and again when maintenance tasks start. This is done to alert the user to maintenance tasks that might be running as well as to report any status messages. There are situations where this display may not be desired, and this can be controlled via several methods.

The **MGUIArgs** Managed Variable can be used to control the interface operation. This is the recommended method for controlling the user interface. Full detail for controlling the display is described in the User Interface Configuration section later in this User Guide. This will limit when the basic interface is displayed and when messages can be displayed.

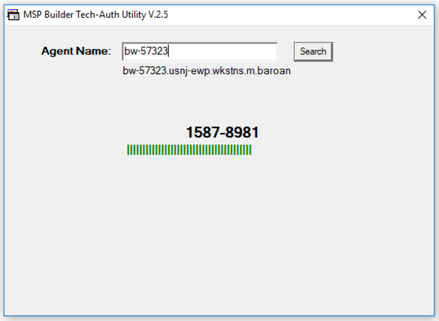
The interface, as of Version 2.5i, supports additional customization features, including language-specific messages, a Security ID, and the ability for the user to generate a support request.

Multi-Language Suport

The installation is provided with a file named **MGUI_en-US.lng.txt**. This file can be edited to replace the English phrases with those of any language. The file should then be saved as **MGUI_*ll-RR*.lng,** where "ll-RR" is the language and region identifier. This identifier can be determined by examining the **HKEY_CURRENT_USER\Control Panel\International : LocaleName** registry value. All language selections are user-specific and default to English if no language file is present.

Security ID

A Security ID is displayed in the center of the User Interface. This pair of 4-digit values is unique to each agent and changes every 60-seconds. This ID can be used when calling the end user to authoritatively identify the MSP technician. The technician has a companion tool that is locked to the MSP's agent and machine-group and can generate the same key when entering the end-user's computer name.



MSP Builder Tech-Auth Utility V.2.5

Agent Name: bw-57323
bw-57323.usnj-ewp.wkstns.m.baroan

1587-8981

The primary purpose of this is to prevent social engineering of the end user by a bad actor impersonating the MSP support staff and asking the end user to go to a website that would potentially install malware. End users should be taught to terminate any calls when this security code cannot be provided or does not match.

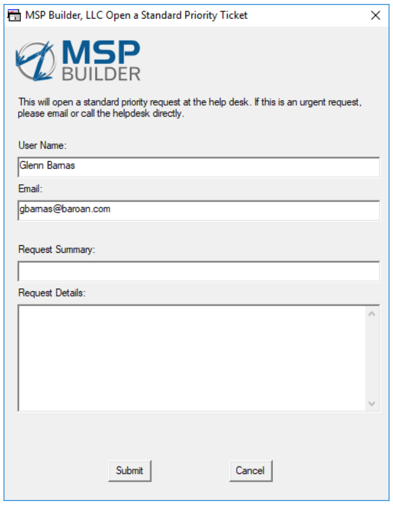
The MSP utility is shown here – the Security Code is displayed when the agent name is entered. A count-down display will show green for 40 seconds, orange when 20 seconds remain, and red when only 10 seconds remain before a new Security ID is generated. If multiple agents match the entered name, the technician is shown all matching entries and prompted to select the correct host.

Support Ticket Request

A “Ticket” button can optionally be displayed on the user interface. The button can be controlled by adding the line “EnableTicketSubmit=Y” to the COMMON section of the MSP_Identity.ini file. When the user clicks the Ticket button, the form shown here is displayed. Once the user enters their name and email, it will be remembered for future requests.

When the Submit button is clicked, the information in the display, plus some information about the user and computer are written to an event that generates an Error Event. This will be processed by the Intelligent Ticket Processing system and sent directly to the MSP’s PSA.

Tickets submitted through this interface will be treated as a standard priority event.



MSP Builder, LLC Open a Standard Priority Ticket

This will open a standard priority request at the help desk. If this is an urgent request, please email or call the helpdesk directly.

User Name: Glenn Bamas

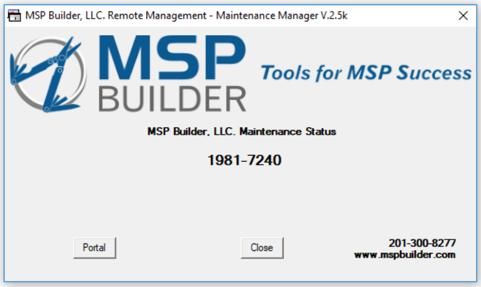
Email: gbamas@baroan.com

Request Summary:

Request Details:

Basic Interface

The basic interface (shown here) allows an RMM Suite user to leverage the Security ID and Portal access even when they do not deploy EMM tools to their customers. The basic interface displays when the agent is in an unmanaged group.



MSP Builder, LLC. Remote Management - Maintenance Manager V.2.5k

MSP Builder, LLC. Maintenance Status

1981-7240

201-300-8277
www.mspbuilder.com

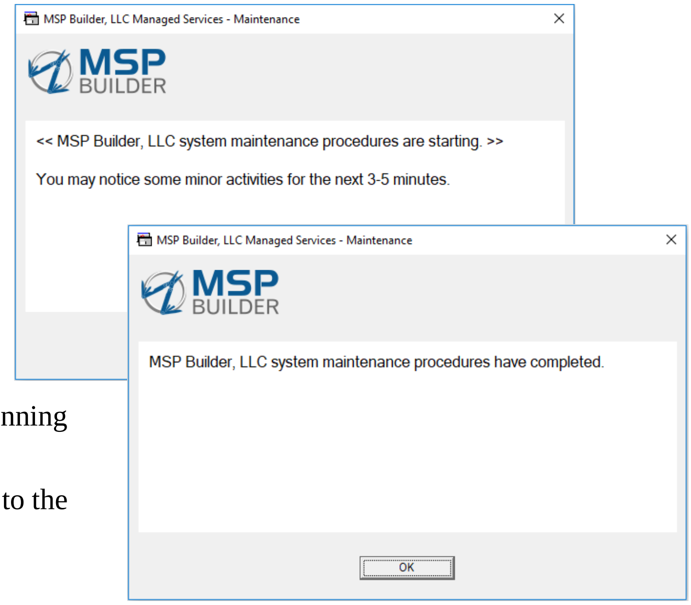
General Maintenance Operation

Kaseya is used to initiate the Sequencer on a daily basis, usually at a random time. To disperse the load, Kaseya initiates daily maintenance between 6 and 8am for servers, and between 10am and 4pm for workstations. This daytime task self-schedules the more intense maintenance tasks to be run after business hours, independent of the VSA. The evening hours default to 7pm for workstations and 8:30pm for servers, but this can be changed by the configuration file.

Reporting status back to the VSA is done through a standardized set of Application Event Log messages. All messages use RMM-MAINTENANCE as the event source, and the Sequencer and each maintenance task have a unique range of Event IDs. Success messages are written to the event log as Information events, but these events are generally not reported back to the VSA.

DAY vs. NITE Maintenance

When the Sequence Engine is called without arguments, it runs in "nightly" mode. In this mode, it is assumed that no user is actively using the system and more invasive tasks that could impact performance or operations can be performed. Nightly maintenance is restricted to non-business hours, and the tasks are configured in the M-SCHEDULE-NITE section. Daytime maintenance requires a --D argument to run, and it is assumed that the tasks configured in the M-SCHEDULE-DAY section will not interfere with a logged-on user or affect the system performance. These tasks should be restricted to data logging or configuration checks. There are no restrictions as to when Daytime maintenance tasks can be run. The user is notified when maintenance starts and completes.


MSP Builder, LLC Managed Services - Maintenance
MSP BUILDER
MSP Builder, LLC Managed Services - Maintenance
MSP BUILDER
OK

<< MSP Builder, LLC system maintenance procedures are starting. >>

You may notice some minor activities for the next 3-5 minutes.

MSP Builder, LLC system maintenance procedures have completed.

One advantage of this is to run Daytime maintenance tasks to gather data during the day AND have it schedule the nightly maintenance via the SCHEDULE command. This will allow the nightly maintenance to start when the computer is running even if not connected to a managed network.

Note that both the logo and the message are specific to the MSP and will not reference “MSP Builder, LLC”.

Utility File Names

The MSP Builder supplied application file names employ a standard format that help identify their purpose and origin. The first 3 characters of the scripts are one of “RMA”, “RMM”, “RMS”, or “RMU”. This is followed by 3 or 4 characters to identify the purpose of the script. All scripts utilize a “.BMS” file extension.

RMA files represent the various utilities that perform various system Audit tasks.

RMM files are generally used directly from the sequencer to perform regular maintenance tasks. They can be invoked manually with alternate command-line arguments to perform checks and enable diagnostic logging modes, or to force the task to run interactively.

RMS files are Smart Monitors and perform intelligent monitoring of the endpoint.

RMU files are considered to be “utility” scripts. This can be invoked via the maintenance sequencer, but are more commonly used as either an interactive process invoked by a technician at the command prompt of the local machine, or as a remediation process through a Kaseya procedure.

There are two special prefix types – **RME** and **RMX**. These are used for specialized tasks and are not run directly by applications or users.

The .BMS file extension is the default type used by MSP Builder and represents a “Builder Management Script”. The RMM Sequencer can invoke *any* type of executable or script, so .EXE, .BAT, .VBS, and .PS# are all valid extensions in addition to .BMS.

Logging

Most maintenance utilities will write logs to summarize the progress or report errors. Logs are written to the KWorking\Logs folder. The log files will include a "_#-DAY" component, where "#" is 0-6 representing SUN to SAT. The leading digit provides an automatic sorting by date of the files. The log files are overwritten each time the maintenance tasks run, so 1 weeks' worth of log data is maintained with no additional management required. If longer term logging is desired, the MSP Builder Log Management tool can be used to create archives on a schedule. The user is also free to develop their own log management tool to move the files to a central or other local path.

Authorization

The RMM Sequencer and many of the utility scripts need license authorization to operate. When the Sequence Engine starts, it checks a local cache file and then connects to the MSP-Builder authorization site via HTTPS protocol to obtain an authorization code if necessary. Codes are issued daily and are good for 14 days to account for when users are traveling. On average, the utilities only need to authorize once every 10 days. When the cache file becomes 10 days old, the utility tries to renew the authorization for another 14 days. The same authorization cache file is used for all EMM Suite utilities, so the first tool to authorize satisfies the license requirement for all other tools during this period.

Command-Line Arguments

The Sequence Engine accepts the following command-line arguments:

--d Run the Sequence Engine in “DAY” mode. This is used to schedule the evening (NITE) maintenance and perform non-invasive procedures, such as checks or data collection.

--i Run the Sequence Engine in "NITE" mode immediately, without any time-of-day restrictions. This is usually invoked via a procedure in response to the user requesting a "Run Now" operation through the user interface (SysTray) tool.

--x*m* Run in Extended Debugging Mode – “*m*” is either “D” for Day mode or “N” for Night mode. Level-2 Debugging is enabled. The requirement for authorization is bypassed. No commands are actually executed, but all logs and event log messages are written. This is intended to perform a manually invoked test/verification of the maintenance procedures.

Core Maintenance Configuration Alarms

The Daily Maintenance sequence engine will generate alarms for two common configuration errors:

1. The agent has been placed directly into the root group. This is unsupported and can result in unexpected actions performed against the agent. This alarm cannot be disabled.
2. The agent has been moved into a different group. This alarm is optional and must be enabled in the configuration file to function.

Maintenance Configuration Files

The maintenance configuration files employ a standard INI file format that can be modified by any text editor. All of the configuration files used by the maintenance utilities are kept in the KWorking folder. The standard configuration files are synchronized with the target system every time the maintenance procedure is deployed. Prior to copying the files, all "RMM_*.ini" files will be removed to ensure the latest configuration files are deployed from the Kaseya server.

Customized configurations can be deployed by creating an appropriate subfolder in the \CustCfg folder and populating the folder with the desired configuration file. After synchronizing the standard configuration files, Kaseya will check for a <customer_id> folder, copying those files to the KWorking folder. If a <group>.<customer_id> folder exists, it will copy those files to the target, overwriting any standard configuration files. It then checks for a <computer_name>.<group>.<customer_id> folder, copying any files it finds to the target, again overwriting any previously written files. This allows a combination of generic, company-specific, group/site specific, and host-specific configuration files to be maintained centrally.

A technician may duplicate any INI configuration file, renaming the file extension to ".INF". This will create a local file that overrides any centrally managed configuration file. This should be done only for testing or configuring a custom version that will eventually become a centrally managed INI file. Note that any locally defined host-specific files (RMM_<type>_<host>.inf) must be deleted manually if they are no longer required. *Use of the .INF file format should be used judiciously to avoid unusual and difficult to diagnose situations where maintenance tasks do not run as expected.*

Configuration File Location

The configuration files are stored on the Kaseya server, in the subfolders within the \Kaseya\WebPages\ManagedFiles\VSASharedFiles\RMM_Maint_CustCfg folder. The "VSASharedFiles" folder is often shared to allow technicians access to these folders to maintain files that are deployed to endpoint managed systems. The standard subfolders that are part of the RMM Suite are:

_Common	Contains the configuration files that are common to all platforms, including the configuration files used by most of the maintenance tools and utilities.
_Servers	Contains the default RMM_Maintenance configuration file for server platforms.
_Workstations	Contains the default RMM_Maintenance configuration file for workstation platforms.

All of the custom folders are maintained here as well. To create a custom workstation maintenance plan for the customer "standardoil", first create the folder "standardoil_Workstations". Copy the default file from the _Workstations folder to the new folder, edit the contents as needed. Don't forget to customize the first comment to indicate the customer and reason for customization!

Configuration File Format

The Maintenance.ini file consists of several sections that taken together, define that tasks that will be performed during a maintenance cycle. Each standard and custom section will be reviewed below.

MAINTENANCE

This is a general configuration section and the parameters here control how maintenance will run.

- Debug= Y|N Optional Default: False
  This enables debugging of the maintenance process. When Debug is enabled (T), additional information is written to the logs, and many maintenance tasks will not actually execute.
- GrpChangeAlm=Y|N Optional Default: False/disabled
  This enabled an alarm to be triggered when an agent moves from one group to another.
- MTime=HH:MM Optional Default: 7 PM (workstations) 8 PM
  The time that evening (NITE) maintenance tasks should start. This value MUST be later than the Exclude value’s second parameter.
- Exclude=am,pm Optional Default: 7.5,19 (07:30 to 19:00)
  Specifies the start and end time when NITE cycle tasks will not be permitted to run. This should generally run from 7 AM or so through 4:30 PM, allowing NITE tasks to be scheduled close to 5 PM.
- MinVer=#.# Optional Default: 6.0 (Vista/Server 2008)
  The minimum O/S version where maintenance should be run.
- Aging1=#,Aging2=# Optional Default: 2, 4
  These define the number of days before maintenance warning messages are displayed if maintenance has not run for the specified number of days. Note that weekends are not included if the value is “2”.

M-SCHEDULE-DAY

This section defines the maintenance tasks and their schedules. These tasks run during the day at whatever time Kaseya randomly deploys the task. This time range is between 10 AM and 4 PM for workstations, and 6-8 AM for servers. The tasks are configured as a *NAME=Schedule* set, where each NAME defines a unique task section later in the configuration file. “NAME” must exactly match the task section, while “Schedule” should be one of the following options:

- Daily – run every day
- Every *day* – run every week on the specified weekday. “day” is the complete day name, such as “Tuesday”, although capitalization is ignored. A special day name of “random” is permitted to distribute tasks throughout the network. Every agent is assigned a random number between 1 and 7, which represents it’s “random day of the week”. This should be used when a maintenance task generates network traffic, such as uploading report data.
- *Specific Day* – this runs a task once monthly on the specific weekday. The values for “specific” can be “first”, “second”, “third”, “fourth”, or “last”. “Day” is the weekday name.

Tasks configured for the DAY cycle should be chosen not to affect the user performance or interrupt their desktop experience.

M-SCHEDULE-NITE

This section is configured exactly as above to define tasks that run in the evening (after business hours). The only difference is that these tasks begin at the time specified by the MTime parameter. The tasks scheduled here should be those that might interrupt the user or affect their performance.

TASKNAME

One or more sections that define the tasks to run. Each section must have a unique name, and the name should describe the task being performed, such as “LOCAL_BACKUP” or “CHECKDISK”. The name must match the name defined in the M-SCHEDULE sections, and should not contain spaces. There are several parameters, required and optional, in each task section.

- **Command=*command*   Required   Default: none**
  Specify the name of the command to run. This should be the EXE or BAT file to execute for Shell methods, or the internal command name. Internal commands include are listed here with their required argument syntax, if any:
  - **Message** – Display a text message on the User GUI
    Arguments=timeout;title_message;body_message
    Note: Semicolon field delimiter! Allows use of commas in message body.
    - Timeout – number of seconds before the display times-out and closes.
    - Title_Message – text for the window title bar.
    - Body_Message – the text message to display. Supports "/n" for newline.
  - **Message2** – Display a text message ***or*** graphic image and customized button text.
    Arguments=timeout;button_type;delay_seconds;btn_text;title_message;body_message
    - Timeout – number of seconds before the display times-out and closes.
    - Button_Type – 0 is "OK", 9 is a custom text message.
    - Delay_Seconds – 0 to 10 seconds delay before the button is enabled.
    - Btn_Text – The text to display in the button.
    - Title_Message – text for the window title bar.
    - Body_Message – the text message to display OR an image descriptor ("IMG:file.bmp"). Only BMP format images are supported, and must be 550 x 200 pixels in size.
  - **Reboot** – Reboot the agent (with notification and ability to defer up to 6 hours).
    Arguments=Not Used **-** An optional special parameter "UptimeControl" is used.
    UptimeControl=**#**
    - UptimeControl is a numeric value that, when defined, can prevent the reboot if the uptime is less than the UptimeCheck warning days *plus* the number of days specified by the UptimeControl value. For example, if UptimeControl is "4" and the UptimeCheck warning value is "15", the reboot will be performed only if the uptime is greater than 19 days (Warning Value + UptimeControl value). This allows the user to respond to the UptimeCheck warning twice and reboot at their convenience before the system is restarted automatically.
  - **UptimeCheck** – perform a check of the system uptime and alert the user if it is excessive.
    Arguments=warn_days,alarm_days
  - **Schedule** – pass the arguments to the SCHTASKS.EXE utility.
    Arguments=Schtasks command line
  - **RegWrite** – write or delete a registry value. Will delete if only Key_Path and Value are defined.
    Arguments=key_path,value,data,type
  - **Download**=URL,Save_As_Filespec
    Arguments=http(s)://full.url,D:\path\file.type
  - **Delete**=Filespec
    Arguments=D:\path\file.type

  When running MSP Builder BMS scripts, the Command must be "RMMKSE.EXE".
- **Arguments   Optional   Default: none**
  The arguments that are passed to the command. For MSP Builder BMS Scripts, the script name and any arguments are defined here.
- **Method   Required   Default: none**
  Can be blank for internal commands, but otherwise is required for all external commands. For

most purposes, this will be “Shell”, which runs the command in a system shell and waits for it to complete. The other option is “RUN”, which runs the command in a system shell, but does not wait for it to complete. This is available for special situations and is generally not recommended.

- **LogInfo** **Optional** **Default: none**
  A brief (less than 50 character) message that is written to the log file to summarize the daily maintenance tasks.
- **Platform** **Optional** **Default: none**
  Restricts the task to 32 or 64-bit platforms. Specify “x86” for 32-bit or “x64” for 64-bit platforms. This option is usually used in task pairs to perform a single task with 32 or 64-bit specific commands. Useful for driver or special application updates.
- **HostControl** **Optional** **Default: none**
  Restricts the task to RDS or HPV (Hyper-V) systems, or using a “-“ prefix, prevents the task from running on these host types.
- **PrimaryDayOnly** **Optional** **Default: off**
  When this value is enabled, the defined task can only run on its defined day and is not eligible for execution when the scheduler is running with Catch-Up enabled. This setting is appropriate for reminder messages that are day-specific.
- **Control** **Optional** **Default: none**
  This can restrict the conditions under which the task is executed. The following methods are supported:
  - FILE;Iflag;Path_To_File
    This will run the command only when the file defined in “Path_To_File” is present.
  - RKEY;Iflag;Path_To_Key
    This will run the command only when the registry path defined in “Path_To_Key” is present.
  - RVAL;Iflag;value_to_match;Key;Value_Name
    This will run the command only when the value in the defined registry Key & Value_Name matches the “value_to_match”.
  - ROLE;Iflag;Role_ID(s)
    This will run the command only when the Role_ID specified matches one of the system roles identified by the daily audit. Often used to perform role/app-specific tasks. If multiple roles are defined, the task will run if any role is found. If the Role ID is prefixed with a “+”, that role is required and can be used to perform tasks when multiple roles are found, such as only when DHCP is found on a Domain Controller. The roles matched are defined by the RMM Suite Daily Audit tool.

    If multiple Roles are defined, they must be delimited with commas.

IMPORTANT: The “Iflag” value will reverse the operation if it is true (1).

WIN-MB Maint - <Type> Procedure Overview

This set of procedures is used to deploy regular maintenance each day to all subscribed endpoints. The procedure is scheduled for execution during the day and internally schedules the NITE tasks. All of the procedures in this collection are essentially the same, the only difference being the *type* of configuration file being deployed – Server or Workstation. Additionally, the Workstation procedure stops, updates, and restarts the User Interface (unless suppressed).

Procedure Outline

This is the generic process followed by all maintenance procedures. Step 9 is specific to Workstation maintenance and (re)starts the User Maintenance system tray interface.

1. The most current MSP Builder tools are deployed/updated by the Daily Tasks procedure.
2. **Workstations Only –** The User Interface management is performed:
    a. The User Interface is stopped by command. If the interface persists beyond 20 seconds, it is forcibly terminated.
    b. The RMMUMA.BAT file is removed from the Startup folder, if present.
    c. If the MGUIArgs Managed Variable *does not contain "*--X", the latest RMMUMA.BAT file is copied to the Startup folder.
    d. The RMMUMA.BAT file is executed – optionally with the arguments defined in MGUIArgs – to restart the User Interface. Unless the "--U" argument is provided, the User Interface displays to report status for about 6 seconds. This application is run in the User context so that the user can interact with the utility to make requests and open tickets. The console is displayed for 6 seconds after starting and can be opened via an icon on the System Tray.
3. The configuration files are replaced with copies from the RMM server.
    a. All .INI files are removed from the KWorking folder.
    b. A "simple" RMM_Maintenance.ini file containing only the MSP identity data is copied.
    c. For all platforms, the _Common folder contents are copied to the KWorking folder.
    d. The platform-specific set of default configuration files are copied to the KWorking folder.
    e. If the folder <Customer>_TYPE is found, its contents are copied to the KWorking folder, overwriting any previous configuration files. This delivers client-specific configuration files.
    f. If the folder <Customer>_TYPE is found, its contents are copied to the KWorking folder, overwriting any previous configuration files. This delivers client site-specific configuration files.
    g. If the folder <machine>.<group>.<Customer> is found, its contents are copied to the KWorking folder, overwriting any previous configuration files. This delivers host-specific configuration files. Note that this does not require the _TYPE suffix.
4. The Data Query procedure is scheduled for 20-minutes in the future, allowing data from Maintenance and the Smart Monitors to update the Daily Audit data collection.
5. The RMMRSP.bat file is executed, specifying the desired Kworking folder, the RMMSEQ.BMS application, and the "--d" argument to run DAY-mode maintenance. This mode performs data collection and then schedules the more intrusive maintenance procedures for after working hours.

Creating a Custom Configuration File

The standard maintenance may need to be customized for customers. The most common customization tasks are:

- Preventing a task from running
- Changing the schedule of a specific task
- Changing when the evening maintenance starts (default is 7pm)

- Changing the "DAY" time range (default is 7:30am to 7pm)

This procedure can also be used to deploy a custom configuration file used by a maintenance script

Start by mapping a drive to the VSASharedFiles folder.

Browse to the RMM_Maint_CustCfg folder. The default folders for maintenance configuration are _Workstations, _Servers, _Hypervisor_Hosts, and _RDS_Hosts. Most often, you will be dealing with workstations, so be sure to use the correct folder!

Create a subfolder for the host, group, or customer that will hold the custom config file. The folder must be properly named!

- For all computers of a specific type at a customer (all sites), use the name <GroupID>_Type, where <GroupID> is the Kaseya customer ID (the part of the name before the ".root. name").
- For all computers of a specific type at a specific customer site, use the name <MachineGroupID>_Type, where <MachingGroupID> is the complete Kaseya customer machine group name (site.type.m.customer format).
- For a specific computer, use the complete host.site.type.m.customer name format.

Custom Configuration File

Create a custom configuration file for a specific maintenance task in the folder created above. See the specific maintenance task document page for information on the custom configuration file format and content.

Custom Maintenance Task or Schedule

Copy the default RMM_Maintenance.ini from the source folder to your new folder. If you are targeting a workstation, that would be the file in the "_Workstations" folder.
BE SURE TO COPY, NOT MOVE!

Edit the RMM_Maintenance.ini file that you just copied.

- Change the comment on the first line to indicate that it is customized for a specific situation.
- If necessary, change the scheduled items. Focus on items in the M-SCHEDULE_NITE section, as these are the more invasive tasks. The items in M-SCHEDULE_DAY should not usually be changed without consulting with the NOC Management team (with the exception of the UPTIMECK process, which alerts to excessive uptime).
    - You can prevent a task from running by setting the schedule to "Never", by placing a ";" in front of the line, or deleting the entire line. (Deleting the line is not recommended.)
    - Changing the schedule is done by simply redefining the schedule value. These values include:
        - "Never" – skip this scheduled event
        - "Daily" – run every day
        - "Every DAY" – run every "DAY" ("DAY" = Sunday through Saturday)
        - "WEEK DAY" – run on the specified week/day of the month.
          ("WEEK" = first, second, third, fourth, or last; "DAY" = Sunday through Saturday)
- Optionally, change the evening start time by removing the comment from the "MTime=" value, and then set the desired evening start time. The time defined MUST be 23:59 or less – ***you cannot schedule the start time for after midnight unless the DAY tasks start after midnight*** (this requires a special maintenance policy to schedule the maintenance at this non-standard time).
- Optionally, change the NITE Exclusion time range by uncommenting the Exclude= line and defining the range. These are decimal numbers, not time values, so 06:30 is represented by "6.5".

If you need a new procedure to be run, it requires a definition in the appropriate M-SCHEDULE section, and a procedure definition in a custom-named section. Any task can be configured to run by creating a definition in the M-SCHEDULE section to define the schedule, and a procedure section to define what to run, how to execute the command, and what arguments to use.

RMMCKD – SMART Disk Check

This utility performs a SMART check and reports any errors detected. It then runs CHKDSK and checks for reported errors. If errors were reported, the user is notified of the errors and that a repair will be attempted at midnight, requiring a reboot.

Configuration Options

There are two sections in the configuration file – CHKDSK and SMART ATTRIBUTES.

The CHKDSK section controls the operation of the RMMCKD utility.

- Drives A comma-delimited list of disks to check. Default is All Disks.
- DoChkDsk A Y/N control to run the Windows ChkDsk command
- DoSmart A Y/N control to run the S.M.A.R.T. disk analysis.

The SMART ATTRIBUTES section maps the response codes to specific error messages. This section should not be modified.

RMMLSB – Local System Backup

This utility will copy key user files to an alternate location for quick recovery when the user profile has become corrupt. File types include Favorites, Shortcuts, Templates, and the printer configurations.

Configuration Options

There are several sections to control general operation and the configuration of each backup topic.

The COMMON section:

- **Debug** Extra information is logged when enabled.
- **DestRoot** Defines the path to the root of the backup folder.
- **SkipSRP** Skips creating the System Restore Point if true.

There are 6 sections that control the operation of the available backup options. Each section has the following parameters:

- **Method** How the task operates, either SHELL or COPY
- **Command** If method is SHELL, the command to run, with arguments.
- **Source** If method is COPY, the source path to copy files from.

The following sections are available: REGISTRY, TEMPLATES, FAVORITES, SHORTCUTS, NK2FILES, and PRINTERCFG. If a section is not present or has no parameters, the backup of those items is not processed.

RMMSCU – System Cleanup

All temporary file locations are scanned, and eligible files are removed, maintaining the available free space on all disk volumes.

Configuration Options

There is only one configuration section – SETTINGS – in this configuration file:

- **TempDirList** A semi-colon delimited list of folders to process.
  DEFAULT: %SystemDrive%\temp;%SystemDrive%\tmp;%SystemRoot%\temp;D:\temp

- **TempFileAge** The file's age, in days, before it can be deleted.
  DEFAULT: 5
- **HotFixFileAge** The age of hot-fix backup files. Only used on Win-7 and earlier.
  DEFAULT: 90
- **OtherDirList** Additional non-temp folders to clean up.
  DEFAULT: *none*
- **ForceDirList** A list of folders to forcibly clear.
  DEFAULT: C:\Windows\Logs\CBS
- **DoNotClearRecycleBin** Prevents clearing the recycle bin when true.
  DEFAULT: N (Recycle Bin is cleared)

RMMVDU – Volume Defrag Utility

A utility to perform a disk defrag process to maintain performance of the disk subsystem. This utility is "SSD-Aware" and will not target SSDs. It will use the standard Windows defrag tool by default, but can be configured to use an alternate product that supports a command-line interface.

Configuration Options

A single configuration section – COMMON – defines the Volume Defrag options:

- **Services** A comma-delimited list of services to stop / start.
  DEFAULT: *none*
- **PreCmd** A command to run prior to running the defrag.
  DEFAULT: *none*
- **PostCmd** A command to run after running the defrag.
  DEFAULT: *none*
- **Volumes** A comma-delimited list of volumes to defrag
  DEFAULT: all volumes.

Event IDs Assigned to Maintenance Tasks

The Sequencer and each maintenance task is assigned a range of Event IDs. The Event Log messages are defined below, along with the appropriate response to detection of the event. Event IDs ending with zero (except for the sequencer, which uses 100 & 101) generally denote a status message that indicates that the task has run successfully. These are informational messages and are not alerted on.

All event log source values are “RMM_MAINTENANCE”.

Event Type Codes:

0	Success
1	Error
2	Warning
4	Information

Component	Source	Type(s)	Event	Message
RMMSEQ	Sequence Engine	1,2,4	100	RMM-Maintenance Complete-NITE
RMMSEQ	Sequence Engine	1,2,4	101	RMM-Maintenance Complete-DAY
RMMSEQ	Sequence Engine	2	102	RMM-Maintenance – Invalid Task or Configuration Data
RMMSEQ	Sequence Engine	1	104	RMM-Maintenance – Task Failed
RMMSEQ	Sequence Engine	1	105	
RMMSEQ	Sequence Engine	2	105	
RMMSEQ	Sequence Engine	1,2	108	RMM-Maintenance – Excessive Uptime
RMMSCU	Disk Cleanup	4	120	RMMSCU: System Cleanup is complete
RMMVDU	Disk Defrag Utility	4	130	RMMVDU: Defrag process complete
RMMVDU	Disk Defrag Utility	1	131	RMMVDU: Defrag failed
RMMLSB	Local System Backup	4	150	RMMLSB: Backup process is complete
RMMLSB	Local System Backup	4	151	RMMLSB: Backup process completed with errors
RMMLSB	Local System Backup	1	152	RMMLSB: System Restore Point creation failed
RMMLSB	Local System Backup	1	153	RMMLSB: No successful backup in 7+ days [DEPRECATED]
RMMCKD	Disk Health Checks	4	190	RMMCKD: CheckDisk process complete
RMMCKD	Disk Health Checks	1	191	RMMCKD: Chkdsk /R scheduled
RMMCKD	Disk Health Checks	1	192	RMMCKD: Chkdsk reports errors
RMMCKD	Disk Health Checks	1	193	RMMCKD: CheckDisk SMART check - ERROR DETECTED!

RMM Internal Procedures

These procedures are utilized by maintenance, remediation, and other audit tasks that are not generally run as interactive processes. Remediation procedures, in particular, should never be invoked outside of the automation process. While most of these are performed as scheduled tasks or in response to alerts, the NOC Managers should be familiar with these should they need to be run for troubleshooting or when initially deploying the Core Automation Suite.

WIN-Agent Daily Tasks - <type>

The Daily Tasks procedure runs on servers or workstations to perform regularly scheduled tasks, including:

- Deploy the current MSP_Identity file, which controls debug status.
- Removes the prior audit data file (SysInfo.ini) to prepare for the current day's audit.
- Executes daily maintenance tasks (if enabled). Separate and unique configuration files are deployed for servers and workstations.
- Initiates the Smart Monitors process via the Smart Monitor Sequencer, if enabled:
    - AV/AM Status Check (workstations)
    - Server Boot During Business Hours
    - Safe Mode Detection (servers)
    - Server Operational Checks (*future release*)
    - Disk Capacity Checks
    - User & Group Security Checks
    - Time Sync Validation
    - Server Performance Checks (*future release*)
- Performs the Daily Audit process.

Note that the "<Type>" procedure sets the type and then invokes the **WIN-Agent Daily Tasks** procedure to perform all of the listed tasks.

WIN-MB Maint

This procedure starts by deploying all of the standard maintenance scripts and support applications. Any existing configuration files are also deleted at this time, ensuring that only the latest copies are present when maintenance runs. The configuration files are then deployed in a least-to-most specific manner.

- The generic configuration files for the system type are deployed.
- Configuration files specific to the customer, if present, are deployed, overwriting any prior files.
- Configuration files specific to the customer's machine group, if present, are deployed, overwriting any prior files.
- Configuration files specific to the computer, if present, are deployed, overwriting any prior files.

The Data Query procedure is scheduled to run 20 minutes after maintenance starts. This will collect the data gathered during maintenance and update the custom fields for the machine. When maintenance is not used, the Data Query procedure is run immediately from the Daily Tasks procedure.

If the user interface is running (workstations only), it is terminated and a new instance started. This ensures that the latest version of the user interface is running. This is a "soft" termination – the interface is commanded to shut down gracefully. After 30 seconds, if it is still running, it is forcibly terminated.

The maintenance sequencer is started, initiating the daily maintenance tasks.

The maintenance process, including configuration file format and all maintenance procedures, is documented fully in the next section.

WIN-MB Data Query – Collect & WIN-MB Data Query – Update Kaseya

These two procedures work in tandem to gather dynamic data from the agent and update custom audit fields in Kaseya. The daily tasks remove the data file to ensure all data is collected anew.

If Maintenance is used, the **Data Query** process runs after maintenance tasks, as some of these utilities will update the audit data file with status information, including AV product and status information. If maintenance is not used, the Data Query process is invoked directly from the Daily Tasks procedure. The **Data Query** process collects data from various sources and updates the working data file. It then uses a mapping file that will extract data from the working file into a data cache file, which maps the information to specific custom fields. The **Data Query – Update Kaseya** procedure is then invoked to extract the information from this data cache and place it into the machine's custom data fields.

Both of the above procedures utilize the RMUSDU.BMS program to manipulate the data in the working and data files. The **Data Query – Update Kaseya** is used only when API Updating is disabled. API-based updating is the default and recommended for optimal performance. Disabling the API updating will force the use of **Data Query – Update Kaseya**, which can add several minutes to the audit process.

System Data Utility (RMASDU.BMS)

The System Data Utility is used to gather data to update standard or custom fields within Kaseya. The utility executes the SYSTEMINFO.EXE program (available on Windows 7 and up) and formats the output into an INI file - SysInfo.ini. The utility then parses a configuration file to obtain data that will be used by Kaseya. The SysDataQuery.ini file contains a list of parameters that will be extracted from various sources, including the SysInfo.ini file. This data will be placed into the SystemData.ini file where it will be ready for the queries made to update the Kaseya data fields.

Command-Line Arguments

--q:name - Extracts the named value from the data cache file and returns it to Kaseya.

<none> - Performs queries and creates the data cache file.

--f - Forces operation on unsupported platforms (XP/2K3 and older) with possible reduced accuracy.

--d - Enables debugging.

The utility functions with two different modes. Run without arguments, it generates the two data files - SysInfo.ini and SystemData.ini. When called with --Q:ValueName, it looks up the data associated with "ValueName" from the SystemData.ini and displays it. This output is captured by Kaseya in a system variable, where it can be used in any procedure, usually to update data fields.

In typical use, a Kaseya procedure might first invoke:

```
cmd.exe /c #KWork#\bin\RMMKSE.exe #KWork#\bin\RMUSDU.BMS
```

This will collect all of the data and place it in the SystemData.ini file (a data cache file), ready to be collected without delay.

When this process completes, pairs of commands will be used to use the command to extract and then update a Kaseya SystemInfo value, as shown here:

```
cmd.exe /c #KWork#\bin\RMMKSE.exe #KWork#\bin\RMUSDU.BMS --Q:AssetTag

updateSystemInfo("Asset Tag", "#global:cmdresults#", All Operating Systems, Continue on Fail)
```

Note that for clarity, the `#KWork#` value above is used to represent `#vAgentConfiguration.AgentTempDir#`.

These two lines will be repeated to extract the various data values that were collected. Any time a new value must be collected, the Data Query procedure must be updated with the Query/Update pair of commands, and the SysDataQuery.ini file must have the value added.

Configuration

The utility employs a configuration file that identifies the values to place into the SystemData.ini file. The values can be obtained from the registry, the SysInfo.ini file, or another INI-format file. The configuration file has one section, called SYSTEMDATAQUERY, and any number of values. Each value consists of 3-4 parameters, delimited with semicolons:

TYPE Either REG, INI, RI, or SI, which defines a Registry or INI/INF file read. "SI" and "RI" extract data from the SysInfo.ini file, which is generated each time the script runs and obtains information about the system. "SI" references data from SystemInfo.exe, while "RI" references data generated by RMM tools. Both types of data are in the same file, but in different sections.

Data1 The registry path or INI file name

Data2 The registry value name or INI file section name

Data3 INI file data value name

An example of the configuration file is shown below:

```
[SYSTEMDATAQUERY]
AssetTag=REG;HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\LanmanServer\Parameters;srvcomment
LastMaint=REG;HKLM\SOFTWARE\RMM\Maintenance;LastMaintRan
LastRestorePoint=REG;HKLM\SOFTWARE\RMM\Maintenance;LastSRPDate
TwainType=INI;%SYSTEMROOT%twain.ini;CAPABILITY;Msg_Get
ComputerDescription=REG;HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\LanmanServer\Parameters;srvcomment
IEVersion=REG;HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer;svcVersion
InstallDate=SI;Original Install Date
```

When the utility runs, it collects the data from the defined location and adds the result to the SystemData.ini file using the same identifier. The identifier should not contain spaces. This effectively caches the data that will be uploaded to Kaseya.

When creating new values, the custom field should be added to Kaseya first. The SystemData.ini file is then updated, followed by the "Data Query" procedure.

User Interface Customization

The Maintenance User Interface is a small GUI application that displays the status of the Daily Maintenance tasks. This interface shows the MSP's logo and contact information, along with the maintenance schedule and a list of the last 50 maintenance tasks. It is also responsible for displaying user information and status messages.

The interface displays briefly when the user logs in, and again when the Daily Maintenance tool starts during the day. The interface minimizes to the System Tray after 5 seconds, or when the [X] or Close button is clicked.

There are several ways to customize the end-user maintenance interface.

Images

There are three images used by the User Interface. For the MSP, these are located in the MSPB_Core\Root\Base_MSP folder, along with the MSP_Identity.INI file and the RMUAIB.INI file.

- RMM_Masthead.BMP – this 550x110 pixel image is displayed across the top of the main interface screen. This image usually contains the MSP's logo and often includes a message or tagline. To match the rest of the interface, the background should be 0xf0f0f0, which is a pale gray color.
- RMM_Logo.BMP – This image is 150x60 pixels, and should also use a 0xf0f0f0 border color to blend into the rest of the interface. This image should contain only the MSP's logo as it is often too small to contain legible text. This logo is displayed on secondary interface screens.
- RMM.ICO – a 32x32 pixel image in ICO format that is used to represent the interface when it is minimized to the System Tray. This image should be easy to distinguish from the Kaseya Agent icon that you choose. The MSP Builder logo is used by default when this image isn't provided by the MSP.

Customers that use the MSP's VSA platform can have customized images that display the customer's logo, or a combination of customer logo and "powered by" MSP logo. To create a customer-specific interface:

1. Create an identity code for the customer of 3-4 characters.
2. Create a subfolder named with the identity code in the MSPB_Core\Root\Base\ folder.
3. Copy the MSP_Identity.INI and RMUAIB.INI files from the _MSP folder to the new subfolder.
    a. Edit the MSP_Identity.INI file, changing the MSPData line to contain the customer's contact information in fields 2, 3, and 4 (Name, Phone, and Web URL). The first and last fields *must not* be modified!
    b. There is no need to modify the RMUAIB.INI file.
4. Define the Identity Code in the MSPID Managed Variable for the customer. When this variable is defined, the value it contains is matched to this folder name and the corresponding images and configuration files are deployed instead of those located in the _MSP folder.

Content

The MSP_Identity.INI file has one configurable setting – the MSPData value. This contains 6 parameters as defined here, delimited with semicolons.

- MSP ID – this is assigned by MSP Builder as a type of User ID. Do not change this field or the RMM Suite tools will be unable to obtain authorized license keys.
- **Company Name** – the name displayed in the title bar and immediately below the masthead image.

- **Phone Number** – usually the main number or the direct help-desk number. Displayed in the lower-left corner.
- **Website URL** – the link to the company's website, displayed below the phone number.
- **Portal URL** – the link to the help-desk ticketing portal. This is usually the MSP's portal, but can be changed if the customer has their own internal helpdesk portal. This defines the action of the Portal button on the bottom row of the interface.

Display Control

As noted above, the interface displays briefly during user login and when maintenance starts each day. This action may need to be suppressed for certain users or platform types, such as POS systems. There are several control methods available.

The brief display of the maintenance interface can be suppressed for an organization or site by setting the MGUIArgs Managed Variable to use one of the optional parameters below. This will not otherwise affect maintenance operations. This affects all agents within a customer organization or site.

Optional Parameters

The MGUIArgs Managed Variable controls operation of the User Interface:

- --U Update Silently; interface displays at login only, and all messages display.
- --Q Operate silently; no interface display, only messages are displayed.
  The interface WILL display if maintenance has not been run in several days.
- --QQ Operate silently; no interface display, only messages are displayed.
  The interface WILL NOT display if maintenance has not been run in several days.
- --K Operate in Kiosk mode – no messages are displayed but the GUI runs on the SysTray.
- --X Suppress User Interface – the GUI is not run, and no messages display.

If the MGUIArgs Managed Variable isn't defined, all User Interface functions will operate/display.

Machine-Specific Override

The interface can be configured on an agent-by-agent basis using a local registry setting. This is meant to be used to override the common settings and should not be used to configure entire environments. This parameter is set using the **WIN-MB Maint – Set GuiSilentStart Flag**. These values override the values provided on the command line. When prompted, enter one of the options from the above list.

Scheduling

Note that it is not possible to control when maintenance runs during the day, thus there is no control over when messages are displayed. The scheduling of the DAY maintenance is controlled by the Kaseya VSA, which randomly initiates the Maintenance process between 10 AM and 4 PM each day. This is done to minimize load on VSA as well as customer resources.

Unless the MSP utilizes aggressive NITE cycle maintenance tasks, the NITE cycle should be scheduled for 16:45 to ensure that it runs each day. A warning will be displayed when the NITE cycle has not run for 3 or more consecutive days, and the user will be allowed to "catch up" on the missed tasks.

Customization

Both the Maintenance and Smart Monitor components share a common customization method. Maintenance has a list of tasks to perform, so there are server and workstation-specific versions of these files. The utilities themselves have configuration files that match the utility name, but with an INI extension. These options are defined earlier in this document for each utility.

All customization should be performed on the VSA server, never on the agent itself! Every configuration file is removed and replaced during the daily tasks to ensure that the latest settings are defined.

Global vs. Specific Configurations

The configuration files provided during the MSP Builder RMM Suite installation are designed to fit most MSP and customer needs. If the MSP wishes to add to (or remove) some of these standard tasks, they should update the files located in the _Common, _Servers, or _Workstations folders. (Only Maintenance utilizes the _Servers and _Workstations subfolders.)

To create a configuration specific to a customer, machine-group, or agent, follow the instructions below to create a subfolder, copy the standard configuration file, and then customize it.

Maintenance

The sections that follow are specific to configuring Daily Maintenance, which use distinct Server and Workstation configuration settings in addition to application specific settings.

Add or Remove Maintenance Tasks

Each task consists of two distinct parts in the RMM_Maintenance.INI file. The first part is the schedule definition, which associates a task with a schedule. The second part is the Task definition, which defines what will be run, and how it is run.

The Schedule definition may be placed in either the M-SCHEDULE_DAY or M-SCHEDULE_NITE section, depending on when you want it to be executed. The format is "TASKNAME=Schedule". "TASKNAME" must match the name of a Task section in the INI file. "Schedule" can be one of:

- Daily – run every day
- Every *day* – Runs weekly on the named day. "Day" can be "MON" through "SUN" and uses the standard 3-character English day abbreviations.
- *Nth day* – runs monthly on the named day in the specified week. Values for the week are First, Second, Third, Fourth, and Last. "Last" can be the fourth or fifth named day of the month.

A special day name called "random" will schedule the task for a random day. This is not random per cycle but per agent. The random schedule day is defined during agent initialization, so a given agent's random schedule is always the same, but will be different than other agents for the same task.

To add a task, a new task section must be defined in the configuration file. This section will define the task command, arguments, and other control parameters. The Task section is illustrated here:

```
[NEW COMMAND]
Command=<name_of_command>
Arguments=<Arguments to pass to Command>
Method=SHELL           (other arguments are available but reserved for special situations)
Control=<parameter>    (Refer to the User Guide for this option)
LogInfo=<Brief description of task for summary log>
Platform=HPV | RDS     (Servers Only - restrict task to Hyper-V or RDS hosts)
```

The provided configuration files are heavily commented and describe the operation and configuration options in great detail. The configuration file should always be the point of reference for current configuration options as the configuration file is updated more often than this manual.

To remove a task, you can either comment out the schedule definition (prefix it with a semicolon) or delete the schedule definition and the task definition. Note that if you delete the task definition but do not remove or disable the schedule definition, an Event 104 error will be generated to report the bad configuration.

Change the NITE Scheduled Time

The evening maintenance start time is defined by the MTime parameter in the MAINTENANCE section. To select a new time, simply enter the time using a 24-hour format.

When a new NITE schedule time is defined, it is important to check the Exclude time parameters. If the NITE time is within the Exclude range, the NITE tasks will never run, and Maintenance will always report failures to run. The Exclude arguments use a decimal time notation to define the start and end of the exclusion period. (The exclusion period prevents tasks scheduled for after hours from running during the day when a maintenance restart is initiated.)

Create a Custom App Configuration

The process outlined here applies to Smart Monitors and Daily Maintenance. Only the folder where the customization occurs is unique. For Daily Maintenance, the configuration files are stored in the MSPB_EMM\Maint_Cfg folder, while Smart Monitor configuration files are stored in the MSPB_EMM\SMon_Cfg folder.

Maintenance – All Customer Servers or Workstations

1. Determine the customer org ID
2. Navigate to the Maint_Cfg folder and create either the “*orgid*_servers” or “*orgid*_workstations” folder. Be sure to replace “orgid” with the proper org name!
3. Copy the RMM_Maintenance.INI file from the _Servers or _Workstations folder, as appropriate for the type of customized file you want to deploy, into the newly created folder.
4. Edit the file you just copied and *immediately* make a note on the first line to define which customer this file is being customized for, the tech that performed the customization, the date of the customization, and the reason for customization. This information will become important for later troubleshooting, so do it *right now!*
5. Make the changes to the file and save it.

Any Group or Machine-Specific Customization

1. Determine the Machine.Group ID. Make sure that the format is *machine.group.orgid* because some data sources represent this as *orgid.group.machine*.
2. Create a subfolder that represents the group or specific machine that you want to deploy the custom configuration to.
3. Copy the configuration.INI file from the _Common, _Servers, or _Workstations folder, as appropriate for the type of customized file you want to deploy, into the newly created folder.
4. Edit the file you just copied and *immediately* make a note on the first line to define which customer this file is being customized for, the tech that performed the customization, the date of the customization, and the reason for customization. This information will become important for later troubleshooting, so do it *right now!*
5. Make the changes to the file and save it.

Once the customized folder is defined, the next time that either Maintenance or Smart Monitors run, it will deploy the default file, then try to copy the contents of a group-specific folder and then a machine-specific folder to the agent. This ensures that the most-specific configuration file is always delivered to an agent. Note that there is no facility to determine if these files exist on the VSA, so a “blind copy” method is used. This will result in procedure errors being logged noting that the files failed to copy. This occurs because the files aren’t present, and this does not indicate a failure of the Maintenance or Smart Monitors.

EMM Suite Utilities

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

RMMSEQ.BMS	Maintenance	The Daily Maintenance “Sequence Engine” – controls maintenance tasks.
RMMSCU.BMS	Maintenance	System Folder Cleanup Utility – performs advanced temp file removal.
RMMVDU.BMS	Maintenance	Volume Defrag Utility – performs disk defrag using built-in or add-on defrag tools.
RMMLSB.BMS	Maintenance	Local System Backup Utility – creates a local backup of key user files.
RMMCKD.BMS	Maintenance	Disk Quality Check Utility – runs SMART and ChkDsk tests.
RMSSSC.BMS	Smart Monitor	System Security Check – validates and updates antivirus settings.
RMSUSC.BMS	Smart Monitor	User Security Monitor - Reports changes to users and groups.
RMSSBM.BMS	Smart Monitor	Server Boot Monitor – detects booting into Safe Mode and server reboots during business hours.
RMSICC.BMS	Smart Monitor	Internet Connection Check – reports when an Internet connection fails over to a backup link or returns to the primary link.
RMSDCC.BMS	Smart Monitor	Disk Capacity Check – defines custom thresholds for every disk volume, self-remediates by running the maintenance tool.
RMSNTP.BMS	Smart Monitor	Network Time Protocol Check – validates the Windows Time Service configuration, checks the local time against both the domain and public time sources.
RMESTM.BMS	Utility	Sys-Tray Monitor – User Interface to Maintenance
RMUW32.BMS	Utility	Windows Time repair utility.