The RMM Suite
Operations & Customization Guide

Daily Maintenance

MSP Builder, LLC
Version 3.0 / Release 22-175
Glenn Barnas

Last Updated: 2023/10/30

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

Contents

Introduction

Regular proactive maintenance is essential to the health and efficient operation of customer computers. The MSP Builder RMM Suite includes a comprehensive Daily Maintenance tool that can run a selection of standard as well as custom task on a daily, weekly, or monthly basis. Further, these tasks can be configured to depend on local computer settings, such as installed software/versions, system Roles or Features, Services, and even specific file and registry data. Like all RMM Suite tools, the configuration that defines the tasks to be run can be customized by customer, location, or even a specific agent.

Overview of Daily Tasks Operation

Daily Maintenance is part of the ***Daily Tasks*** process that the RMM Suite performs on every endpoint. These tasks begin with an audit of key components that can be used to control which tasks are allowed to run. This pre-maintenance audit is an integral part of the RMM Suite's ability to quickly detect and remediate zero-day vulnerabilities with a minimum of effort. This audit does not update the RMM platform - its results are stored locally for use by the other RMM Suite tools.

Once the audit completes, the *Daily Maintenance Sequence Engine* runs. This application is the heart of Daily Maintenance - it decides which tasks are eligible to run, executes them, and manages the results. The maintenance tasks can update the Audit Data Cache file located on the endpoint.

When the maintenance tasks complete, the *Smart Monitor Sequence Engine* starts. Similar to the maintenance sequencer, it examines the platform and decides which of the RMM Suite's Smart Monitors should be run. Some Smart Monitors run once; others run continuously throughout the day. Many Smart Monitors will also update the Audit Data Cache file located on the endpoint.

After the Smart Monitors are initiated, the Daily Audit is executed again, this time collecting additional information not needed for Maintenance operation. This data is added to the Audit Data Cache file, from where data can be read and written to the RMM platforms agent data fields or to documentation engines such as IT Glue and Hudu.

Overview of Daily Maintenance

Daily Maintenance can perform any conceivable operation required to upgrade, maintain, or configure an endpoint computer. A comprehensive set of tools are built-in directly to the sequence engine. These are augmented with additional RMM Suite maintenance apps designed to perform specific tasks. The flexibility of the Daily Maintenance is underscored by its ability to run any external tool, including Batch files, PowerShell scripts, and executables. The flexibility is such that creating RMM-based scripts and configuring deployment schedules can easily be replaced with maintenance tasks.

Built-In Commands

These commands are available directly in the Sequence Engine and do not require any additional files or overhead to run. NOTE: commands marked (*) are not available in the RMM Essentials package.

- SHELL — Execute a program and wait for completion.
- USHELL — Execute a program in the current user's context & continue immediately.
- RUN — Execute a program and continue immediately with the next task.
- RMMSCRIPT* — Execute an RMM-based script. (RMMSCRIPT:<ScriptName>)
- REBOOT — Reboot the computer. The user is given several chances to delay.
- SCHEDULE — Define a Windows Scheduled Task.
- LAUSER* — Create or update the LAUser Account
- UPTIMECHECK — Verify uptime and perform notifications or alarms.
- MESSAGE — Display a text-based message via the Maintenance User Interface.

- MESSAGE2 Display a BMP image message via the Maintenance User Interface.
- DELETE Delete a file.
- REGREAD* Read a registry value into 1 of 5 maintenance variables.
- REGWRITE Create, set, or delete a registry value.
- INIREAD* Read an INI file value into 1 of 5 maintenance variables.
- INIWRITE* Create, modify, or delete an INI file key-value pair.
- DOWNLOAD* Download a file from the specified URL to a specified location.
- UNZIP* Extract the contents of a Zip file to a specified folder location.

The RegRead and IniRead commands can load a variable with data that can be used as arguments in other maintenance tasks. The values persist until overwritten, or the maintenance sequencer app completes its defined tasks.

These commands will be discussed in detail in a later section of this guide.

RMM Suite Maintenance Applications

These applications are part of the RMM Suite and have been specifically designed to perform proactive maintenance tasks and are normally initiated by the Maintenance Sequence Engine.

- RMMSCU System Cleanup - cleans TEMP folders, recycle bins, and other locations.
- RMMVDU Volume Disk Defrag - locates and defrags spinning media.
- RMMLBU Local Backup - copies key user profile data to a second local folder.
- RMMCKD Check Disk Health - runs and analyzes SMART and CHKDSK results.

Advanced Capabilities

Some of the advanced capabilities of Daily Maintenance include:

Role or Application-Specific Tasks

Need to perform application-specific maintenance? The tasks can be defined so they run only when Roles or Applications are installed, such as SQL, Exchange, or even SQL 2016 or QuickBooks 2021. Trigger application backups, updates, or data defrag operations without the need to configure individual systems!

Zero-Day Vulnerability Remediation

When you need to detect where a vulnerable application is installed and update it quickly, Daily Maintenance has you covered. Simply define an audit setting to identify the vulnerable application. Then configure one or more maintenance tasks to initiate the remediation on every vulnerable endpoint. Maintenance can:

- Find the vulnerability that the Daily Audit detected.
- Download a 32 or 64-bit application, based on the endpoint O/S.
- Unzip the download package, if necessary.
- Execute the remediation package

Client-Specific Deployments

Need to deploy an application to all customer workstations, or update an application from a specific current version? Instead of writing an RMM script and figuring out where to run it, simply configure a Daily Maintenance task to run daily. It can detect that the application is missing (or outdated) and install or update it. Once the application is installed and up-to-date, the task will be ignored on those endpoints. The benefit of this method is that as new endpoints are deployed for the customer, these applications will be installed/updated without any additional effort.

Delayed Execution

Many of the RMM Suite Maintenance functions support a Delayed Execution option, allowing the task to be delayed until a specific time later in the day. The time specified must be prior to midnight of the current day, and the computer must be powered on for the task to run. Delayed Execution is useful for tasks that could impact the end-user, such as rebooting the computer or running CPU-intensive tasks.

Local Configuration Files

Throughout this guide, we will illustrate the configuration of the tools using INI files. It is important to understand that all configuration tasks must be done through the Configuration Interface on the mspbuilder.com website. The interface will represent the configuration data in a format similar to the INI files. This data will be delivered to the endpoint via a secure, encrypted data stream and loaded into memory, where it is used by the applications. Each of the RMM Suite tools writes a copy of their configuration data to the local computer in INI-file format to aid in troubleshooting configuration issues. These files will contain a comment on the first line indicating the level from which the data was obtained - Default, Organization, Site, or Agent.

When tools are run individually (not part of the Daily Tasks set), they will use the local configuration file located on the agent. This allows the commands to be run locally to test configuration options by directly editing the config file. These changes should then be defined via the web management portal. *All local configuration files are removed daily at the start of the Daily Tasks cycle - any local changes will be lost.*

Operation

Operation of Daily Maintenance is mostly automatic. The RMM platform is responsible for initiating a Daily Tasks script. The RMM platform usually performs this only on “managed” endpoints, and each RMM platform has a unique method for distinguishing between managed and unmanaged endpoints. From the perspective of the RMM Suite, we use the following definition of Managed and Unmanaged:

Managed Endpoint - the customer expects the endpoint to be fully monitored and proactively maintained, including patching and operating system updates.

Unmanaged Endpoint - the customer is one of two types:

- Break/Fix - no proactive operations are performed on the endpoints unless the customer explicitly requests support. The RMM Suite will generally perform audits on these systems, but not perform any actions that would modify the endpoint beyond installing the management tools. No maintenance, monitoring, patching, or updating is performed.
- New Managed Client in Onboarding Mode - This endpoint belongs to a customer that will eventually be fully managed but may be in the discovery/audit phase where the MSP is still reviewing the environment and preparing for any client-specific requirements. The RMM Suite can assist by collecting necessary audit data, but as above, will not make any significant changes to the endpoint. The MSP can convert the customer to Managed and allow full management to take effect.

Configuration Data Deployment Process

When the RMM Suite is deployed to a client, a default set of maintenance tasks is defined. These represent the most commonly-used tasks for servers and workstations. Only a very few commands are configured for servers, including Temp Cleanup, Uptime Status, and Disk Health. Workstations have some additional tasks configured, including user profile data backup (copy to an alternate folder, retain 7 days/copies) and the display of reminder messages for weekly patching.

By using the RMM Suite Web Management Interface, the MSP / IT team can modify these default actions as well as add custom tasks. The RMM platform syncs asset data regularly, so the configuration parameters can be defined to override the system defaults at a client, client-site, or agent-specific level.

Timing Considerations for Configuration Changes

Making configuration changes in the web management interface will immediately save the configuration data to a master database. These changes are replicated to the cloud servers within 15-minutes, allowing endpoints to obtain data from the regional server closest to the endpoint. Servers are available in the US, UK, EU, and APAC.

Configuration data is deployed to the endpoint once every day. The endpoint requests the update when the Daily Tasks are initiated by the RMM platform. As the schedule for daily tasks is well-defined, changes to configuration should be made at least 30-minutes *before* the start of any schedule, or any time after the last endpoint in the environment runs its daily tasks. If configuration changes are made during the business day, it is assured that some systems will have already received and used the prior settings, while any endpoints due to run will obtain the updated data.

The RMM Suite customer is responsible for ensuring that changes are deployed in a way that provides for operational consistency in their environment.

Configuration of the task parameters is covered in detail in the next section of this guide.

Suppressing Maintenance

There may be times where proactive maintenance is not desired, either in an entire customer environment, a customer site, or a specific endpoint system. The RMM Suite can accommodate this easily through Policy Control.

Policy Control is a mechanism used in the RMM Suite to disable tasks that are otherwise performed automatically. This can control all automation, maintenance, patching, specific monitor sets, and even individual Smart Monitors.

Policy Control is managed on an Org, Site, or Agent Endpoint basis. There are no global defaults for policy control. To suppress the Daily Maintenance tasks entirely, simply select “Maintenance” in the policy control settings for the org, site, or endpoint using the MSP Builder Web Management Interface.

Configuration

The Daily Maintenance uses a single INI format file for operation. The file is called **RMM_Maintenance.INI** and is located in the standard RMM Suite root folder - %PROGRAMDATA%\MSPB. This file is deleted and downloaded every day prior to running the maintenance tasks to ensure that the latest configuration settings are always used. The configuration data is set using the web management interface at www.mspbuilder.com. Access to the configuration interface requires a login to the MSP Builder website; "customer" level access with association with a specific MSP Builder customer, and Data Management level access. This allows our customers to define which of their employees have the ability to manage this configuration data. The configuration data is replicated to an Azure CDN server, ensuring that the endpoint always obtains its configuration data from the closest available server.

When the data is saved, it includes a comment that identifies whether the data are the default settings, or overrides associated with the customer, location, or the specific endpoint. This helps when troubleshooting to identify whether the standard or custom data are being deployed.

Use and operation of the MSP Builder web management interface are covered in a separate guide. This document will focus on the details of the configuration of the maintenance tasks themselves. Note that these examples may contain comments to provide additional clarity. These comments will not exist in the actual configuration files.

The Maintenance Task

Each maintenance task consists of a *Task Record*. Any number of Task Records can be defined to accommodate the maintenance requirements of an MSP or IT Department using the RMM platform for endpoint management.

A typical entry contains the following settings:

```
[TASK-ID]
Target=All | Workstation | Server
Schedule=Daily | Every DAY | Ordinal DAY
PrimaryDayOnly=Y | N
Platform=X86 | X64
DelayUntil=HH:MM
Command=Internal, External, or custom command
Arguments=string of arguments passed to the command
Method=Internal | SHELL | USHELL | RUN
LogInfo=Short Action-Log message
Control=Control where task may run by file or registry data
Roles=list of roles, features, services, or applications where the task may run
```

Not all parameters are necessary - the required parameters are shown in **bold**. The default action is to immediately run the task on all targets and platforms. The optional parameters can be used to control if, where, and when a task is executed. We'll examine each of the lines of the Task Entry.

[TASK-ID]
This is a unique task identifier. Each Task ID must be unique, and any duplicate sections will be ignored - only the first section will be used. This name allows spaces and can be mixed case, but special punctuation characters are NOT allowed. All upper-case is recommended as this makes the Task ID stand-out when reviewing the file.

When similar tasks are required but need different settings for servers and workstations, we recommend using the same name with a suffix to distinguish the Server and Workstation task - [TASK-S] and [TASK-W] represent the same "TASK" but unique options for Server or Workstation platforms.

Target
This can restrict which type of target platform the task is allowed to run on. The default is all platforms, and this can be specified as “All” for clarity. Other options are “Workstation” and “Server”. It should be noted that only the first character of this value is checked, and a single character may be defined. We do recommend that the full term be spelled-out to avoid any ambiguity when reading the information.

Schedule
This defines when the task will be allowed to run. The following options are available:

- **Never** - the task is disabled. This allows the task to remain defined but in a disabled state.
- **Once** - The task is run once as if it had been configured as a Daily task but will be flagged to prevent it from running a second time. This has replaced the Control “Once” in prior versions of the Daily Maintenance tool.
- **Daily** - The task runs every day.
- **Every *DAY*** - The task runs weekly on the defined day. “DAY” is the first three letters of the English day names - SUN, MON, TUE, WED, THU, FRI, and SAT.
- ***Ordinal DAY*** - The task runs monthly on the prescribed DAY, which is the same list as above. “Ordinal” can be “First”, “Second”, “Third”, “Fourth”, or “Last”. The term “Last” will run on either the fourth or fifth occurrence of a given day, depending on how many of those days are in the current calendar month.

PrimaryDayOnly
When maintenance tasks are missed due to a computer being powered off, the tasks that were missed during the prior seven days are eligible to be run the next time maintenance starts. In some cases, it is inappropriate to run in this manner. For example, a message reminding the user that the computer will be patched and rebooted that evening should only display when that externally managed operation is active. Setting PrimaryDayOnly to a true value (T,Y,ON,1) will prevent that task from running except on its assigned day. This has no effect when the schedule is “Daily”.

Platform
This can be used to restrict operations to specific O/S platform types. The options are either “X86” (32-bit) or “X64” (64-bit) and are based on the O/S installed, not the underlying hardware. The task will run on any platform if this is not specified.

This option is especially useful when downloading an update package from the web. A specific package can be downloaded using two almost identical tasks, except one configured to download the 32-bit package on X86 platforms and the 64-bit package on others. If both tasks are configured to download the appropriate source file to a common name, a third task can execute that package file knowing it is already platform-appropriate.

DelayUntil
Maintenance tasks are executed immediately in the sequence they are defined. If a task will place a significant performance load on the endpoint, it can be scheduled to run at a specific later time. The time MUST occur in the current 24-hour day and initiate prior to 23:59. Since maintenance tasks are randomly scheduled by the RMM platform, this delay time must start *after* the last possible maintenance execution time for a given platform (typically 6-8 AM for servers, 10AM to 4 PM for workstations).

Not all task types allow the DelayUntil option. Currently, only SHELL-based commands, Reboot, and Messages can be delayed. When the DelayUntil is specified, the task is scheduled and the next maintenance task (if any) is processed. This has the effect of operating as a RUN method command.

Command
This is the name of the command to execute. For INTERNAL commands, including RMM Suite maintenance apps, it must exactly define the command ID without any additional parameters or path info. For custom commands, the entire path to the command should be defined. For PowerShell cmdlets, the command should be "PowerShell.exe" and the script name (with path) and all other parameters should be placed in the Arguments parameter.

Arguments
The arguments passed to the command. For PowerShell and other scripted languages, the script name should be listed here as the first argument, followed by any other required or optional arguments. Be sure to quote spaces in arguments as necessary! Note that some of the RMM Suite maintenance tools use a comma-delimited list of arguments that must be defined in a specific sequence.

Applies to Both Command and Arguments Parameters:

- The use of Environment Variables is supported.
- The use of Maintenance Sequencer Macros is supported. For more information, see Macros at the end of the Configuration section of this guide.

Method
The method of command execution. "**Internal**" defines either a built-in sequencer command or an RMM Suite Maintenance application. This is optional and will default to "Internal" when these commands are found. "**Shell**" will execute the command as a sub-process and waits for it to complete. The exit code of the command will be returned and used to report the success or failure of the task. "**UShell**" will execute a command in the current user's context via the Maintenance Interface. If no user is active or the User Interface isn't running, the request will be ignored. "**Run**" will execute the command as a unique, detached process and then immediately continue to the next task, if any. Unless the initial execution reports an error (start failure, not found), the task will be reported as successful regardless of how it completes. The RMM Suite customer is responsible for proper logging and determination of any success or failure status of commands initiated via the RUN method.

LogInfo
Each task can optionally report a short (<60 characters) message to a log file. This file is trimmed to 50 lines at the end of maintenance, and the entire log is displayed in the Maintenance Interface as "Recent Maintenance Tasks". This allows the user to monitor the maintenance actions being performed on their system. This is only appropriate for workstations where the Maintenance Interface runs (it does not run on server platforms).

Tasks that perform non-maintenance operations do not need to be logged - the decision is up to the RMM Suite customer as to what to log or not log.

Control
The Control parameter provides several methods to allow or prevent the execution of the task. This can allow the task to be performed based on the presence/absence of a file, a registry key, or matching/not matching a registry data value. The following Control options are supported:

- **FILE;bINVERT;Filespec**
  - FILE - defines a FILE type control
  - bINVERT - Boolean - reverses the match when true (1)
  - Filespec - Text String - the complete file path value

  The task will be allowed to run when the defined file is present and bINVERT is zero, or when the file is not present and bINVERT is one.
- **RKEY;bINVERT;KeyPath**
  - RKEY - defines a Registry Key control
  - bINVERT - Boolean - reverses the match when true (1)

    - KeyPath - the full registry key path - “HKLM\Software\myApp”, for example

  The task will be allowed to run when the defined registry path is present and bINVERT is zero, or when the registry path is not present and bINVERT is one.
- RVAL;bINVERT;MData;KeyPath;Value
    - RVAL - defines a Registry Value type control
    - bINVERT - Boolean - reverses the match when true (1)
    - MData - the data to match in the registry path + value.
    - KeyPath - the full registry key path - “HKLM\Software\myApp”, for example
    - Value - the registry value that contains the data to be compared.

  The task will run when the data in the registry value matches the value defined by MData and bINVERT is zero, or when the data in the registry value does not match the value defined by MData and bINVERT is one.

These parameters provide a significant amount of control over whether a task runs or not. The typical use is to check for a file not present and run an installer; check for a registry to determine if an application key is missing and install the application; and check a registry value to determine the version of an application and decide if it should be updated. Controls play a key role in application installation, maintenance, and Zero-Day remediation tasks.

Roles
The roles parameter is a comma-delimited list of role IDs that are discovered by the RMM Suite Daily Audit and stored in the System Roles field. When any Role ID in the list is matched to the System Roles data, the control is applied.

An example of this parameter will be helpful, so we will assume that Audit has detected SQL 2019 installed on the endpoint. The System Roles data will contain both “SQL” to indicate that SQL Server is installed, and “SQ19” to identify the SQL version as 2019.

- If Roles=SQL, the task will run
- If Roles=SQ16, the task will NOT run, as this control limits the operation to SQL 2016
- If Roles=SQL,-SQ19, the task will run on any version of SQL Server *except* SQL 2019. Note the leading “-“ prefix on the SQ19 parameter, which represents a “not match” operation.
- If Roles=+SQ16,+SQ19, the task will run only where SQL Server 2019 is co-resident with SQL Server 2016. Note the “+” prefix, which represents a “must match” operation.

A special case for controlling tasks via Role Tags is for Class of Service operations. This allows the RMM Suite to perform tasks if the endpoint is assigned to a specific Class of Service. The Role will contain a “sc:*name*” tag that represents the Service Class. Tasks can be configured for more than one service class ID, but these cannot be prefixed with a “+” as an endpoint can only be a member of one service class.

The Class of Service is defined by a Global Custom Field called “CCOS” and is a string value. Any terms can be used to define service class names.

Kaseya VSA: The service class will be set to use the root machine group name (usually “m” or “unm”) if the global (Org) Custom Field “CCOS” is not defined. This functionality is specific to VSA 9.x.

See the **RMM Suite Operations Guide - 12 - Class of Service Operation** document for complete information on leveraging Class of Service for Maintenance and Automated Onboarding.

Internal Commands

The commands available within the RMM Suite Daily Maintenance component are described here. These commands are considered “internal” as they are part of the RMM Suite set of applications. External commands would include custom scripts and applications defined by the RMM Suite customer and executed by the Shell or Run methods.

Most commands support all task controls, but some do not support the DelayUntil or PrimaryDayOnly options. Each command identifies whether these task options are supported.

Shell

Runs a command (EXE, batch, PowerShell, etc.) and waits for the completion (unless DelayUntil is defined). When running a script (VB, PS), the command should initiate the scripting engine (ie: PowerShell.exe) and the Arguments should contain the script name followed by any arguments needed to perform the task.

Supports DelayUntil:	Yes
Supports PrimaryDayOnly:	Yes
Arguments:	Yes - Command-dependent

UShell

Runs a command (EXE, batch, PowerShell, etc.) in the currently logged-in user context. When running a script (VB, PS), the command should initiate the scripting engine (ie: PowerShell.exe) and the Arguments should contain the script name followed by any arguments needed to perform the task. A user must be logged in and the Maintenance User Interface must be running or this task will be ignored. There is no ability to limit this command to a specific user account.

Supports DelayUntil:	No
Supports PrimaryDayOnly:	Yes
Arguments:	Yes - Command-dependent

Run

Runs a command (EXE, batch, PowerShell, etc.) and immediately continues with the next task. When running a script (VB, PS), the command should initiate the scripting engine (ie: PowerShell.exe) and the Arguments should contain the script name followed by any arguments needed to perform the task. This command is not supported in the Essentials package.

Supports DelayUntil:	No
Supports PrimaryDayOnly:	Yes
Arguments:	Yes - Command-dependent

RmmScript:<ScriptName>

Initiates an RMM-based script on the current endpoint. Does NOT wait for completion, reports error only if script was not found or failed to schedule. The complete script name must follow the colon.

Supports DelayUntil:	Yes
Supports PrimaryDayOnly:	Yes
Arguments:	Yes - Command-dependent, optional, “prompt;varname;value” list of comma-delimited value sets. Different RMMs may not use all 3 values, but all 3 are required.

Reboot

Initiates a reboot of the endpoint. If the Maintenance Interface is running, a message will be displayed to the user, and the user will have an option to delay the reboot for 30-minutes. They can initiate this delay up to 4 times for a total of 2 hours. If the user does not respond within 10-minutes, the computer will be rebooted immediately. *Using DelayUntil will only provide a single 15-minute grace period notification.*

Supports DelayUntil:	Yes
Supports PrimaryDayOnly:	Yes
Arguments:	None

Schedule

Schedules a command via Windows Task Scheduler. You must supply all of the SchTasks arguments needed to perform the task. The /F parameter is recommended to force the task creation or deletion.

Supports DelayUntil:	No
Supports PrimaryDayOnly:	Yes
Arguments:	SchTasks command arguments

LAUser

Generates a complex, 16-character password, then updates the “LAUser”* local account with that password. If the account doesn’t exist, it is created, added to the local administrators group, and marked as “hidden” (does not display on the start page account list). The password is then ciphered and written to the system registry. The Daily Audit can be used to read, decipher, and write the password to a custom field or documentation engine password manager to make it available to the support team.

The LAUser account provides a local administrator account with a regularly changing complex password. This account can be used by the MSP for support or given to the end-user under special circumstances (installing drivers while off-network and require UAC) without compromising other accounts or providing elevated access.

When used without an argument, it duplicates the original LAUser functionality and creates an account with the user ID “lauser” (Local Admin User). If an argument is used, it represents an alternate user ID. This must be one word with no spaces. Multiple accounts can be created as necessary. Each will have a unique password

The ciphered password is written to the MSP Builder registry key with a value of “SEC_PHASH_<userid>”. This must be added to the Daily Audit to read this and store it in the appropriate custom field.

Supports DelayUntil:	No
Supports PrimaryDayOnly:	No
Arguments:	None | Username

UptimeCheck

Checks the uptime of the endpoint as reported by the Operating System, then compares the uptime to the Warn# and Alarm# values. If the uptime exceeds these values, and the bWarn or bAlarm options are enabled, an RMM alert event will be triggered. If bNotify is enabled, the user will be notified of excessive uptime of their computer with a suggestion to reboot at their earliest convenience. The reboot is not enforced. The notice is displayed every 3 days until a reboot occurs or an alarm is generated in the RMM.

The bNotify is not supported on servers and will be ignored. Employing the alarm option on servers will help identify systems that have not rebooted during monthly patching.

Supports DelayUntil: No

Supports PrimaryDayOnly: No

Arguments: Warn#,Alarm#,bWarn,bAlarm,bNotify

Warn#	Number of days for a warning condition
Alarm#	Number of days for an alarm condition
bWarn	Y|N, allows an warning to be triggered if true
bAlarm	Y|N, allows an alarm to be triggered if true
bNotify	Y|N, Reminds the user to reboot after Warn# days

Recommended options:

Servers	30,45,N,Y,N
Other	15,35,N,Y,Y

Message

Displays a text-based message via the Maintenance Interface. The message will be displayed with an [OK] button and will self-close when the timeout expires. Messages force a System Modal operation with the message box initially displayed on top of all other applications.

Supports DelayUntil: Yes*
*Requires latest version of the Maintenance Interface

Supports PrimaryDayOnly: Yes

Arguments: Timeout;Title;Message Text

Timeout	Time, in seconds, before the message self-closes
Title	The text for the message pop-up box title
Message Text	The message to display - use “\n” for line breaks

Message2

Displays a text or image-based message via the Maintenance Interface. The message will be displayed with an [OK] button and will self-close when the timeout expires. Messages force a System Modal operation with the message box initially displayed on top of all other applications.

This Message command provides additional control over the button. The button text can be defined, and the button can be displayed in a disabled state for a specified delay to prevent a user from quickly or accidentally dismissing the message.

Supports DelayUntil: Yes*
*Requires latest version of the Maintenance Interface

Supports PrimaryDayOnly: Yes

Arguments: Timeout;Mode;Delay;BtnTxt;Title;Message

Timeout	Time, in seconds, before the message self-closes
Mode	Button style - must be “9”
Delay	Seconds of delay before button is enabled

BtnTxt	The text inside the button
Title	The text for the message pop-up box title
Message	Text message OR image tag &IMG:ImageFile.bmp& The Image Tag can specify a full path filespec. If the path is omitted, the BMP format image must be located in the MSPB root folder - %PROGRAMDATA%\MSPB.

NOTE: Both message commands will display the message inside of a message area within an overall pop-up dialog box which has the RMM Suite customer logo displayed in the upper-left corner.

NOTE: Both message commands delimit the arguments with semicolons to allow commas in the message text. Most other internal commands use commas to delimit the values.

NOTE: The RMM Suite tools, including the user interface, attempt to update daily. Access to the MSP Builder web URLs must be permitted in any endpoint AV or security tools for updated to be successful.

Delete

Deletes the specified file without warning if it exists. Wildcards are allowed. Folders will not be deleted.

Supports DelayUntil:	No
Supports PrimaryDayOnly:	Yes
Arguments:	The path and file name of the file to delete.

RegRead

Reads the data from the specified registry key:value and saves it to the defined User-Defined Macro variable. There are five user-defined macro variables named V1 through V5. The data stored in these macro variables can be used in the Command and Argument parameter of any task that follows the RegRead command. If multiple RegRead tasks are defined and assigned to the same V# macro variable, the data in that value will be replaced with the new data.

Supports DelayUntil:	No
Supports PrimaryDayOnly:	No
Arguments:	V#,Key,Value **V#** The Macro Variable to assign, from V1 to V5 **Key** The registry key path **Value** The registry value to read

RegWrite

Writes a value to a specified registry location. The registry key path, value, data, and data type must be specified. If the data and data type are not specified, the corresponding value will be deleted.

Supports DelayUntil:	No
Supports PrimaryDayOnly:	No
Arguments:	Key,Value,Data,Type **Key** The registry key path **Value** The registry value to read **Data** The data to be written **Type** The data type to write, such as REG_SZ

IniRead

Reads the data from the specified INI file, section, and value and saves it to the defined User-Defined Macro variable. There are five user-defined macro variables named V1 through V5. The data stored in these macro variables can be used in the Command and Argument parameter of any task that follows the IniRead command. If multiple IniRead tasks are defined and assigned to the same V# macro variable, the data in that value will be replaced with the new data.

Supports DelayUntil: No

Supports PrimaryDayOnly: No

Arguments: V#,File,Section,Value

V#	The Macro Variable to assign, from V1 to V5
File	The complete path filespec of the INI file to read
Section	The section in the INI file to query
Value	The section value to read

IniWrite

Writes a value to the specified INI file, Section, and Value. If the data is not defined, the value is deleted from the INI file

Supports DelayUntil: No

Supports PrimaryDayOnly: No

Arguments: File,Section,Value,Data

File	The complete path filespec of the INI file
Section	The section in the INI file to update
Value	The registry value to write
Data	The data to be written

Download

Downloads a file from the specified URL and saves it in the specified location and name.

Supports DelayUntil: No

Supports PrimaryDayOnly: No

Arguments: DL_URL,SaveToFile

DL_URL	The URL to download the file from
SaveToFile	The full path\filename to save the download to

Unzip

Unzips a standard ZIP file using the command-line Unzip.EXE provided as part of the RMM Suite set of applications. If this file is not present, the Unziup command will not function. You must specify the full path to the Zip file as well as the full path to the directory where the files should be extracted to. The directory will be created if necessary prior to extraction.

Supports DelayUntil: No

Supports PrimaryDayOnly: No

Arguments: ZipFile,ExtractDir

ZipFile	The full path to the Zip file to extract
ExtractDir	The full path to the folder to extract files to

Internal Applications

These are maintenance applications provided with the RMM Suite. As they are external applications controlled by a Maintenance Task, they are considered “internal” but will be supported by an external configuration file that permits much more customization than what is possible by the task control parameters. These generally perform more complex operations, and new applications can easily be introduced in the future.

RMMSCU - Disk Cleanup

The Disk Cleanup app is run daily to maintain the available disk space on the endpoint. The default settings focus on the C: drive but can be configured to maintain space in any drive and folder location. Files in temp locations that have not been modified in the past few days are removed to reclaim space. All standard TEMP locations, the Recycle Bin, and user temp folders are processed. User folders include Temp, Temporary Internet Files, and History, and - optionally - the user’s Downloads folder.

Supports DelayUntil:	Yes
Supports PrimaryDayOnly:	Yes
Arguments:	Not Required - use config file for maintenance tasks
Config File:	RMMSCU.INI
Application File (Command):	RMMSCU.BMS
Alarms:	None

The application does accept the following command-line arguments, but these are intended for when the application is invoked manually. The configuration file should always be used when configuring this task in Daily Maintenance. This allows the task to be placed into the Global level and applied to every client, but custom configuration files deployed to Orgs, Sites, or Agents to provide the customizations needed.

Arguments

--age:#	Override the default file age of 5 days.
--force	Force a cleanup with a minimal age. This is used when there is an extreme low-space condition. This WILL clear the user’s Downloads folders!
--NoRBin	Disables emptying the recycle bin.

Config File

The default settings in the configuration data are the same as the internal application defaults.

```
;RMMSCU.INI - optional file to override default settings in the RMMSCU.BMS app
[SETTINGS]
; comma-delimited list of TEMP folder paths
;  DEFAULT:  %SystemDrive%\temp;%SystemDrive%\tmp;%SystemRoot%\temp;D:\temp
TempDirList=%SystemDrive%\temp;%SystemDrive%\tmp;%SystemRoot%\temp;D:\temp
;
; the default number of days to allow temp files to live (DEFAULT: 5)
TempFileAge=5
;
; comma-delimited list of other directories to clear based on file age
; use this to clean application-specific content
OtherDirList=
;
; comma-delimited list of other directories to FORCE-clear
ForceDirList=C:\Windows\Logs\CBS
```

Continued on next page

```
;
; Disable recycle bin cleanup if "Y"
; - OR set reg HKLM\SOFTWARE\RMM\Maintenance : DoNoClearRecycleBin : 1 : REG_DWORD
DoNotClearRecycleBin=n
;
; Clean the user Downloads folder. A numeric value greater than TempFileAge can be
; specified to allow the downloads to remain on the system for an extended period.
ClearUserDownLoads=30
```

RMMVDU - Volume Defrag Utility

The Volume Defrag app allows the regular execution of a disk defrag tool to help maintain efficient operation on spinning media drives. This app is smart enough to not execute a defrag tool on SSD or flash-based media. The built-in Windows Defrag application is invoked with arguments that direct it to target specific drives.

Supports DelayUntil:	Yes
Supports PrimaryDayOnly:	Yes
Arguments:	--C | --O **--C** targets the System drive, **--O** targets other, non-system volumes. Two separate maintenance tasks are needed to defrag all volumes and should be scheduled for different days.
Config File:	RMMVDU.INI
Application File (Command):	RMMVDU.BMS
Alarms:	13x Series

The application is NOT scheduled by default. It should be enabled in the default configuration only when large quantities of spinning media drives are used.

Config File

The default settings in the configuration data are the same as the internal application defaults.

```
[COMMON]
; Settings common to the defrag process - these are RARELY NEEDED!!
;
; comma-delimited list of services to stop / restart
Services=
; Command to run prior to starting defrag
PreCmd=
; Command to run after completing defrag
PostCmd=
; comma-delimited list of drives to defrag
; ALL local volumes are detected & processed - if only specific volumes
; should be processed, they should be listed here as this list overrides
; the volumes detected.
;Volumes=E:,G:
```

Alarms

The following alarms are reported by this utility:

- 131 - ERROR - Defrag Failed - <VOLID> - <Error Message>

RMMLSB - Local System Backup

This maintenance app will make copies of key user profile data, keeping 7 days' worth of information. While not a substitute for true off-device backups, this will allow a rapid recovery of Favorites, Shortcuts, Templates, and other key profile data in the event of accidental deletion or profile corruption.

Supports DelayUntil:	Yes
Supports PrimaryDayOnly:	Yes
Arguments:	None
Config File:	RMMLSB.INI
Application File (Command):	RMMLSB.BMS
Alarms:	15x series

The application is scheduled to run daily by default. Even though it runs daily, the System Restore Point is performed at least every 7 days.

Config File

The default settings in the configuration data are the same as the internal application defaults.

```
; Configuration file for Local Backup Utility
; Performs external commands or file copy operations
; Macros are available for dynamic path replacement
; &CFGPATH&      - The RMM Suite Working Directory (%PROGRAMDATA%\MSPB\)

[COMMON]
; Define the root folder path where the files are copied to
DestRoot=&CFGPATH&Backup\
; Should we skip creating the System Restore point each week?
SkipSRP=N
; Enable or disable specific backup components - Favorites, Shortcuts, Templates
; or Printer Configuration. DEFAULT value is listed first
BackupFavorites=Y|N
BackupShortcuts=Y|N
BackupTemplates=Y|N
BackupPrinterCfg=N|Y
```

If it is not desired to perform the task, Set the action to "N".

Alarms

The following alarms are reported by this utility:

- 151 - INFO - Backup completed with errors
- 152 - ERROR - System Restore Point Creation failed (error)

RMMCKD - Disk Health Check

The Disk Health Check performs two distinct tests on the endpoint disk drives. The first test queries the drive's SMART data and analyzes the results. Errors are reported to the RMM platform as alarms. The second test utilizes the Windows ChkDsk command. The command is run and the output is parsed for error messages. If the endpoint is a server, the error generates an alarm on the RMM platform. If the endpoint is a workstation, a repair is scheduled for midnight and the user is notified of the need to reboot and repair. If - after the repair attempt completes, errors are still detected, an RMM alarm is triggered.

Supports DelayUntil:	Yes
Supports PrimaryDayOnly:	Yes
Arguments:	None
Config File:	RMMCKD.INI
Application File (Command):	RMMCKD.BMS
Alarms:	19x Series

The application is scheduled to run daily by default. Even though it runs daily, the System Restore Point is performed at least every 7 days.

Config File

The default settings in the configuration data are the same as the internal application defaults.

```
[CHKDSK]
; Define the scope of the checks - All, System Only, or all Except System (Other)
; The default scope is "All"
Scope=All|System|Other
; check only specific drives - default is to identify/check all local drives
; Ignored when Scope=System; When Scope=All, the System drive is added to this list.
; Drives in the list that are not present are ignored without reporting an error.
Drives=
; Individual checks can be disabled. ChkDsk and SMART tests are enabled by default.
DoChkDsk=Y
DoSmart=Y
```

Alarms

The following alarms are reported by this utility:

- 191 - ERROR - Chkdsk /R scheduled on <VOLID>
- 192 - ERROR - Chkdsk reports errors - <VOLID>
- 193 - ERROR - CheckDisk performing SMART check - ERROR DETECTED!

Macros

The maintenance system supports several built-in macros that can be used in the Command and Arguments parameters of a task configuration. Macros are defined by surrounding the macro with an ampersand, thus: “&MACRO&”.

Macros are processed twice for each task, When the task Command and Arguments parameters are read, the standard macro replacements are performed. Then, when the command is executed, the macro replacement is performed a second time, allowing the User Defined macros to be set.

Important: When User Defined Macros are used, if the macro value has not been defined prior to running the task that uses it, the macro tag will be removed prior to executing the task! This is not considered an error condition as these User Defined Macros could contain optional parameters.

&CFGPATH&	The path to the root of the RMM Suite working folder, usually %PROGRAMDATA%\MSPB.
&LCLHOST&	The local computer name.
&ARCH&	The computer’s O/S architecture, either X86 (32-bit) or X64 (64-bit).
&V#&	Up to 5 User-Defined Macros. These can be set via RegRead and IniRead commands. Each time one of the read commands is executed, the result of the read action is stored in the assigned variable V1 through V5. The values will be retained until replaced by another read command using that variable or the sequencer terminates.

Environment Variables

In addition to the supported macros, any environment variable can be used in the Command and Arguments parameters of a task. Environment variables must be wrapped with “%” characters (as usual) to be referenced properly. We strongly recommend the use of environment variables for common system paths instead of hard-coding whenever possible.

Administration

While the actual administration of the Daily Maintenance is done through the Web Management Interface, the *concept* of applying the configuration data is also part of the administration, and that is what will be covered in this section.

There are some important points to understand:

- The Daily Maintenance itself consists of a set of tasks performed on a schedule in a particular sequence. The sequence of tasks can be configured in the Web Management Interface. In particular, tasks that load User Defined Macro variables need to occur before they are referenced, and commands like UNZIP probably need to run after the DOWNLOAD command that obtains the Zip file. It is important to consider the order of operation for certain tasks. Other tasks that do not have Macro or other dependencies can run in any order.
- Multiple levels of configuration are supported by the RMM Suite. The levels apply to an entire configuration set, such as Daily Maintenance Tasks or the Disk Cleanup app. The levels include:
    - Global Settings - configurations that should apply to all endpoints unless they are specifically overridden. These represent the default settings for the RMM Suite customer and are initially provided by MSP Builder during RMM Suite installation.
    - Org Settings - these settings apply to a specific customer in the RMM. These *replace* the entire set of configuration data defined by the Global Settings.
    - Site Settings - some customers will have multiple locations that can be identified in the RMM platform. These settings apply only to endpoints in a specific customer org+site and *replace* settings defined at either the Global or Org levels.
    - Agent Settings - apply to a specific agent and allow “special case” endpoints to be directly targeted. Agent settings will *replace* Global, Org, or Site-level configurations.

To minimize the administration effort and maximize the power of the RMM Suite will require a bit of planning. You should always consider what settings have the broadest application and use those as your platform’s defaults in Global Settings. When an entire customer needs additional tasks or different settings, define Org-Level settings. The same concept applies to configurations that should apply only to one location at a customer, or to a specific machine.

Obviously, creating overrides results in additional administration effort, so overrides should be considered carefully. Always define your configuration at the broadest possible level.

When a new configuration set is created for an org, site, or agent level, the data will be initialized based on any higher level customizations. For example, if you create a special task for a customer and then create a new set of tasks for a specific site for that customer, the new configuration will inherit the special task defined at the Org level. The override can remove that extra task or define additional tasks.

Zero-Day Remediation Process

Daily Maintenance can make it simple to respond to zero-day remediation tasks by creating a series of tasks that leverage specific controls to download and install the update, only when the vulnerable condition is detected. The entire process can be completed in a single maintenance cycle.

1. Start the process by configuring the Daily Audit to search for and report on the vulnerable component. Refer to the **RMM Suite Operations Guide - Daily Audit** for details on configuring this detection option. In short, the audit should identify the vulnerable component and assign it a component ID. Maintenance has the ability using the Roles parameter to match on any Role Component ID.

2. Define the task(s) needed to initiate the remediation. This could be calling a single script that does all the work, or it can be a series of Maintenance tasks. We will examine using Maintenance to perform all remediation tasks.

   Assumptions:
   - An update package needs to be downloaded and installed
   - Separate packages are available for 32 and 64-bit platforms
   - The update needs to be unzipped
   - A file from the Zip package needs to be executed
   - A registry parameter needs to be set to configure an optional setting

     In these examples, only the required parameters will be shown!

3. Create two separate download tasks, one for 32 and one for 64-bit platforms

```
; Download the update for BadApp Version 2.3 - Role ID is "BA23"
; Use the same target name for both platforms - only one will be used
[REMEDIATE_BADAPP_2.3_STEP_1_X86]
Target=All
Schedule=Daily
Platform=X86
Command=Download
Arguments=https:dl.badapps.org/32bUpdate2.4.zip,%TEMP%\UpdateBA.zip
Method=Internal
Roles=BA23

[REMEDIATE_BADAPP_2.3_STEP_1_X64]
Target=All
Schedule=Daily
Platform=X64
Command=Download
Arguments=https:dl.badapps.org/64bUpdate2.4.zip,%TEMP%\UpdateBA.zip
Method=Internal
Roles=BA23
```

4. Unzip the download package to a folder in the Temp folder

```
; Unzip the downloaded update package
[REMEDIATE_BADAPP_2.3_STEP_2]
Target=All
Schedule=Daily
Command=UNZIP
Arguments=%TEMP%\UpdateBA.zip,%TEMP%\UpdateBA\
Method=Internal
Roles=BA23
```

5. Execute the update command “setup.exe” from the extraction folder with the -SILENT argument

```
; Execute the setup.exe with the -SILENT argument
[REMEDIATE_BADAPP_2.3_STEP_3]
Target=All
Schedule=Daily
Command=%TEMP%\UpdateBA\Setup.exe
Arguments=-SILENT
Method=Shell
Roles=BA23
```

6. Write the configuration parameter to the registry

```
; Add the registry setting to disable open access
[REMEDIATE_BADAPP_2.3_STEP_4]
Target=All
Schedule=Daily
Command=RegWrite
Arguments=HKLM\SOFTWARE\BadAppsInc\Settings,Disable Open Access,1,REG_DWORD
Method=Internal
Roles=BA23
```

Once this series of tasks run, Audit will run and no longer detect Bad App Version 2.3. The Roles field will no longer match on “BA23”, preventing these tasks from running in the future. By using the Daily

schedule, any machines offline when the configuration is deployed will update as soon as they come online and run Daily Maintenance.

Appendix I: Configuration Examples

These examples will be shown as they are delivered from the management server. All task parameters are defined, even if no data is present. These parameters are ignored and - where appropriate - will use application default values.

Daily Disk Cleanup

This is a standard task provided by the RMM Suite and runs on all platform types.

```
; Run a Disk Cleanup daily on all systems except Hyper-V hosts
[CLEANUP]
Target=All
Schedule=Daily
PrimaryDayOnly=N
Platform=
DelayUntil=
Command=RMMSCU.BMS
Arguments=
Method=Internal
LogInfo=Temp File Cleanup
Control=
Roles=
```

Disk Defrag

This is a standard task provided by the RMM Suite but is disabled by default. It performs a defrag on spinning-media drives. All SSD media is detected and skipped. There are two tasks - one for the primary disk and one for all secondary disks.

```
; Run a Disk Defrag on the system disk monthly on all systems except Hyper-V hosts
; DISABLED by default - schedule should be something like “Third Friday”
[DEFRAG_SYSTEM]
Target=All
Schedule=Never
PrimaryDayOnly=N
Platform=
DelayUntil=22:30
Command=RMMVDU.BMS
Arguments=--C
Method=Internal
LogInfo=Defrag - SYSTEM Volume
Control=
Roles=

; Run a Disk Defrag on non-system disks monthly on all systems except Hyper-V hosts
; DISABLED by default - schedule should be something like “First Friday”
[DEFRAG_OTHER]
Target=All
Schedule=Never
PrimaryDayOnly=N
Platform=
DelayUntil=22:30
Command=RMMVDU.BMS
Arguments=--O
Method=Internal
LogInfo=Defrag - Non-SYSTEM Volumes
Control=
Roles=
```

Local Profile Backup

This is a standard task provided by the RMM Suite and runs on workstation platform types. It creates a local copy of essential user profile data and maintains 7 distinct daily copies. This is useful for when a user accidentally modifies or deletes a file or their profile becomes corrupt. The backup copies Templates, Favorites, Shortcuts, and Printer Connections.

```
; Local backup of critical profile files
[LOCALBACKUP]
Target=Workstations
Schedule=Daily
PrimaryDayOnly=N
Platform=
DelayUntil=
Command=RMMLSB.BMS
Arguments=
Method=Internal
LogInfo=Local User Data Files backed up
Control=
Roles=
```

Disable Windows Update

This is a standard task provided by the RMM Suite and runs on all platform types. Most RMM platforms use Windows Update to perform Patch Scans. If Windows Update is not turned off after the scan, the computer can update and reboot under control of Windows Update. These updates will likely occur at times other than what are desired. This task simply makes sure that Windows Update is disabled on the morning following Patch Scans.\

```
;Disable Windows Automatic Updates for Windows machines.
[DISABLEWINUPDATE]
Target=All
Schedule=Every Tuesday
PrimaryDayOnly=N
Platform=
DelayUntil=
Command=REGWRITE
Arguments=HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU,NoAu
toUpdate,1,REG_DWORD
Method=Internal
LogInfo=Disable Windows Automatic Updates
Control=
Roles=
```

Take a System Snapshot

This is a standard task provided by the RMM Suite and runs on all platform types. It runs daily and creates a text file containing information about running processes, services, and other system parameters. The data provides a point-in-time view of what is running on a given machine.

```
; Take a process & service snapshot
[SYSSNAP]
Target=All
Schedule=Daily
PrimaryDayOnly=N
Platform=
DelayUntil=
Command=RMUSNAP.BMS
Arguments=
Method=Internal
LogInfo=System Process snapshot
Control=
Roles=
```

Disk Health Check

This is a standard task provided by the RMM Suite and runs on all platform types but uses separate task configurations for servers and workstations. This performs a SMART status analysis, followed by a CHKDSK. The workstation configuration allows the application to notify the user of detected errors and schedule a repair action to run at midnight. The repair requires a reboot, which is why it is suppressed on the server task.

```
; Check disk consistency and SMART errors
; Workstation mode - can force CHKDSK /F and reboot
[CHECKDISK-W]
Target=Workstations
Schedule=Every Wednesday
PrimaryDayOnly=N
Platform=
DelayUntil=
Command=RMMCKD.BMS
Arguments=
Method=Internal
LogInfo=Local Disk Health Checks
Control=
Roles=

; Check disk consistency and SMART errors
; Server mode - Generates alert on CHKDSK failures, will not fix/reboot
[CHECKDISK-S]
Target=Servers
Schedule=Every Wednesday
PrimaryDayOnly=N
Platform=
DelayUntil=
Command=RMMCKD.BMS
Arguments=--S
Method=Internal
LogInfo=Local Disk Health Checks
Control=
Roles=
```

Reboot

This is a standard task provided by the RMM Suite but is disabled by default. It can be used to perform a regular reboot of any computer type, although this example is restricted to workstations.

```
; Perform a regularly scheduled reboot at 11:45 PM unless the
; SkipMaintReboot Registry flag is set.
; DISABLED by default - schedule could be “Last Friday” or “Every Sunday”
[REBOOT]
Target=Workstations
Schedule=Never
PrimaryDayOnly=N
Platform=
DelayUntil=23:45
Command=Reboot
Arguments=
Method=Internal
LogInfo=System was rebooted by maintenance
Control=RVAL;1;1;HKLM\SOFTWARE\RMM\Maintenance;SkipMaintReboot
Roles=
```

Display a Reminder Message for Patch Night

This is a standard task provided by the RMM Suite but is disabled by default. There are two types of messages supported - Text and Image. The Image type allows graphics to be combined with text for a more eye-catching appearance.

This example is has specific content to remind users of upcoming patching schedules. It is NOT part of patching, but simply a method to leverage the Maintenance application to display a reminder message.

Note that the text message contains several "\n" tags to control line-breaks in the message.

Both of these examples have a DelayUntil parameter. This depends on running the latest version of the Maintenance Interface which has the ability to schedule messages. Unlike other tasks that use the Task Scheduler to initiate the command at a specific time, the message includes the time parameter in the message request. If an older Maintenance Interface is used, the message will be displayed immediately upon the task being executed.

```
; Display a text-based message each week
[MESSAGE-PATCH_NITE1]
Target=Workstations
Schedule=Never
PrimaryDayOnly=Y
Platform=
DelayUntil=16:45
Command=Message
Arguments=86400;Tonight is Patch Night!;Important security patches will be installed
tonight.\n\nYour system must be rebooted to complete the patching.\n\nPlease save your
work and log off at the end of your workday\n\n          - - - DO NOT SHUTDOWN! - - -
Method=Internal
LogInfo=Display Patch Night reminder message
Control=
Roles=

; Display an image-based message each week
[MESSAGE-PATCH_NITE2]
Target=Workstations
Schedule=Every Wednesday
PrimaryDayOnly=Y
Platform=
DelayUntil=16:45
Command=Message2
Arguments=86400;9;5;I Acknowledge;Tonight is Patch Night!;&IMG:PatchNotice.bmp&
Method=Internal
LogInfo=Display Patch Night reminder message
Control=
Roles=
```

Note that when the MSPB Patch Utility is used, the Patch Night Reminder will be configured to run DAILY, and will have a control of:
"`RVAL;0;1;HKLM\SOFTWARE\RMM\Maintenance;AllowPatchReminder`".
The patch tool will automatically set and clear the registry value to allow the message to be displayed on the day prior to the patching schedule.

Uptime Check / Notify / Alarm

This is a standard task provided by the RMM Suite and runs on all platform types but uses separate task configurations for servers and workstations. The maintenance app checks the uptime and compares it against the arguments provided. The app can display a notice to the user (workstations only) regarding excessive uptime and recommend a reboot and trigger warning and alarm events to alert the MSP about systems with high uptime values.

```
; ===== NOTE =====
; Argument syntax is:
; WARN (# days),ALARM (# days),WarnEvent (Y/N),AlarmEvent (Y/N),AlertUser (Y/N)
; WARN/ALARM represent the number of days of uptime that define the warning
; and alarm thresholds
; WarnEvent is a Boolean value - when TRUE, will generate Warning Event Log
; entries and corresponding tickets
; AlarmEvent is a Boolean value - when TRUE, will generate an Error Event Log
; entry and corresponding ticket
; AlertUser is a Boolean value - when TRUE, will pop up an alert to the user
; each day in warning state recommending a reboot

; Check workstation the uptime and alert if exceeds the specified number of days
[UPTIMECK-W]
Target=Workstations
Schedule=Daily
PrimaryDayOnly=N
Platform=
DelayUntil=
Command=UptimeCheck
Arguments=15,35,N,Y,Y
Method=Internal
LogInfo=Check System Uptime
Control=
Roles=

; Check the server uptime and alert if exceeds the specified number of days, except
Hyper-V hosts
[UPTIMECK-S]
Target=Servers
Schedule=Daily
PrimaryDayOnly=N
Platform=
DelayUntil=
Command=UptimeCheck
Arguments=30,45,N,Y,N
Method=Internal
LogInfo=Check System Uptime
Control=
Roles=
```

Appendix II: 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	100	RMM-Maintenance: Complete - one or more tasks failed to execute
RMMSEQ	Sequence Engine	2	100	RMM-Maintenance: Complete - maintenance ended
RMMSEQ	Sequence Engine	4	101	RMM-Maintenance: Complete - all tasks completd.
RMMSEQ	Sequence Engine	2	102	RMM-Maintenance: Invalid Task <ID>
RMMSEQ	Sequence Engine	2	102	RMM-Maintenance: Invalid Configuration Data: <message>
RMMSEQ	Sequence Engine	1	104	RMM-Maintenance: Task Failed
RMMSEQ	Sequence Engine	2	105	RMM-Maintenance: WARNING-Agent machine group has changed
RMMSEQ	Sequence Engine	1	105	RMM-Maintenance: ALARM-Agent machine group is invalid: <grp>
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 - <VolID> - <ErrorMessage>
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
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!