The RMM Suite
Operations & Customization Guide

Utility Apps

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

Last Updated: 2025/05/22

MSP Builder RMM Suite

The MSP Builder RMM Suite contains proprietary software, including unpublished source code.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 number of RMM platforms. 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 LLC
385 Falmouth Ave
Elmwood Park, NJ 07407
201-300-8277

Contents

Introduction

The MSP Builder RMM Suite includes a variety of Utility applications that enhance the operation of the RMM platform while also serving to offload a significant amount of processing. These applications allow the use of advanced programming concepts that function well above the capabilities of typical RMM platform scripting languages.

Like our other tools, these applications use a .BMS file extension, and nearly all utilize a 6 or 7 character file name that begins with “RMU”. These tools employ a compiled language to achieve operational speed and enhance platform security. These tools are checked daily - the file timestamp and checksum are both compared to a manifest file and any mismatch will result in the correct version being downloaded from our website.

This Page Intentionally Left Blank.

Use and Operation

Each of the Utility Applications will be described on the following pages using a consistent format. When the utility is initiated by an RMM platform script, that script will be identified.

All RMM Suite tools are located in the PROGRAMDATA\MSPB\Bin folder. Most tools are invoked using the RSRUN.BAT script, located in the Bin subfolder. This batch file accepts the name of the tool (without the .BMS extension), verifies the components are present, and then initiates the command. The syntax of this batch file is

```
%PROGRAMDATA%\MSPB\Bin\RSRUN.BAT <RS_AppName> [<arguments>]
```

Arguments are optional and all arguments passed to the batch file - except for the app name - are passed to the application.

Logs are generated by all applications and are written to the PROGRAMDATA\MSPB\Logs folder. The log files follow a common format, starting with the AppID, optionally followed by a day part, then a ".log" extension. Any application that runs frequently includes the day-part in the log file name, while applications that run rarely or just once will omit this part of the file name. The Day-Part format is #-DAY, where "#" is in the range of 0 (Sunday) through 6 (Saturday). The "DAY" is the 3-letter English abbreviation for the standard day name. This format groups all similar logs together and in day sequence when sorted by name.

When a log file already exists, but has not been modified in the current day, it's contents are overwritten. When the log file has been modified in the current day, it will be appended to if the application is run multiple times in a single day.

RMUAAR - User Admin Rights

Find, Report, and Remove unauthorized local admin accounts

Checks the local administrators group for unauthorized accounts and removes them. By default, only the Administrator account is permitted, although a list of additional accounts can be specified.

Restricted to WORKSTATION-ONLY operation.

Platforms:

Runs on any RMM platform.

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--A	OPTIONAL - Performs an Audit - reports on any non-standard accounts in the Administrators group on the local computer.
--L:list	OPTIONAL - Specify a comma-delimited list of additional allowed local accounts. By default, only "Administrator" account and the "Domain Admins" group are permitted.

Config File

No configuration file is used.

Log File

RMUUAR.log

RMUOBA - Onboarding Automation & Config State Management

Run apps or RMM Scripts for agent initialization and config management

Using a config file, the app runs a series of RMM Suite Apps or RMM Scripts (collectively "tasks") from the endpoint to install and configure new agents. Performs general, Class of Service-based, customer-specific, and/or location-specific tasks. Runs on any endpoint after the first check-in and daily thereafter. When any task is executed, it automatically calls an RMM Script to report the onboarding status. If no tasks are executed then no status script will be run.

Each task successfully installed updates tracking maintained on the local system, and the task cannot be run again through the OBA application unless the FORCE option is manually provided.

The OBA application runs daily as part of the Daily Tasks to maintain the configuration state of the endpoint. Tasks can be defined to uninstall old applications and install new applications as supported products change and evolve. This is referred to as *Configuration State Management*.

Platforms:

Depends on the RMM Platform's ability to initiate and report status of a script via the API. All platforms support direct MSPB BMS app execution for deployment and configuration.

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--D	Enable DEBUG mode
--F	FORCE running all tasks (usually one-time only)
--S	Schedule the task to run and exit. This is the normal execution method when invoked from the RMM platform. This does not need to be used when invoking this application manually.
--U	Skip the install process on unmanaged systems

Config File

RMUOBA.INI - Defines the RMM scripts to be executed. There are multiple sections in the configuration data.

Each section defines the name of an RMM Script or Procedure, *exactly* matching the name in the RMM Platform. The name is followed by either No, Yes, or ALL. These control the execution of the scripts in the following way:

- No — The task will not be run. This allows the procedure to be staged for future use or temporarily disabled without removing it's definition.
- Yes — The task will be run on Managed agents only.
- All — The task will be run on both Managed and Unmanaged agents. Tasks are often suppressed on unmanaged agents, although certain tasks, such as special audits or security scans might be performed on every agent as it checks in for the first time.
- <COS_ID>[;TAG] — The task will be run only if the organization's assigned Class of Service (CCOS Global Custom Field) matches the defined COS_ID string. Note that the COS_ID tag is supported only in the 3 Global Task sections ALL, ALL-SERVERS, and ALL-WKSTNS. Multiple service class IDs can be listed, separated with commas. A TAG can optionally be added to control execution based on matching/not matching a System Roles TAG value. See the **Class of Service Operations Guide** for full information on COS.

Local Commands

OBA/CSM can run local MSP Builder apps directly in addition to RMM platform scripts. This process is much faster with less overhead but is limited to automation supported by the MSPB tools. A key advantage of this method is that additional arguments can be specified to further customize the process.

Instead of defining the RMM script name, define the RMM Suite app name (without the .BMS extension) and arguments using the following syntax:

```
RS:appid [arguments]=control
```

This method is supported for any control and in both the global and client task sections. The most common RMM Suite apps would be **RMUCET (**which requires a single argument to define an action package) and **RMUAMT*** (which runs the MSPB App Management tool to install, update, or remove apps). MSP Builder can create and host custom application packages that can perform many common tasks such as download, unzip, and execute installation packages with specific arguments. Find more information about the RMUCET and RMUAMB apps elsewhere in this document.

*RMUAMT requires the optional Application Management license to operate.

The OBA and CSM actions can be controlled on a per-endpoint basis by applying Policy Control tags.

- XOBA - suppress both OBA and CSM actions. This functions only when the tag is applied at the Organization level as the agent CF cannot be defined before agent installation.
- XCSM - Permit OBA but then suppress the daily CSM operations.

Global Tasks

The following sections define the names of tasks that will be invoked globally:

- ALL Run these procedures on all agents for any organization.
- ALL-WKSTNS Run these procedures on all workstations for any organization.
- ALL-SERVERS Run these procedures on all servers for any organization.

Excluding Tasks

For each of the “ALL” categories, there is a similar section with “_EXCLUDE” suffix. These sections also define the name of an RMM Script or Procedure, but the values are a comma-delimited list of customer org IDs where these global tasks should NOT be run. This may represent a configuration where a specific client either doesn’t want a product installed or uses alternate options when installing.

Whenever possible, RMM Scripts should use Public Variables to define configuration, allowing these values to control how the tasks run, rather than creating custom tasks and exclusions.

Customer Org and Site Tasks

This option illustrates the extreme configuration capabilities of the RMM Suite OBA tool. The following four section types can be used to perform customer-specific tasks, such as installing Line of Business applications or other customer-specific configurations.

- Custid-ALL Run these procedures on all agents for the specified organization.
- Custid -WKSTNS Run these procedures on all workstations for the specified organization.
- Custid -SERVERS Run these procedures on all servers for the specified organization.
- Custid-Wkstns.site Run these procedures on all workstations for the specified organization in the specific site-group.

Sample Config File

Note that most use a “yes” parameter, but the customer-specific security tool has an “all” argument to install on every computer, even if it is unmanaged.

```
; These procedures will be executed on all platforms based on a Class of Service match.
[ALL]
WIN-Webroot - Installation (MV)=Bronze;-WRT
; This runs on all Silver or Gold classes if the S1AV tag is not present
WIN-Sentinel One - Installation (MV)=Silver,Gold;-S1AV

; These procedures will be executed on all server platforms
[ALL-SERVERS]
; Define exclusions for ALL processes

; These procedures will be executed on all workstation platforms
[ALL-WKSTNS]
;WIN-Win-10 Fast Boot - Disable=yes
WIN-Office 365 - Install - 2019 Office Pro Plus=Yes

; CLASS OF SERVICE EXAMPLE
; Install Chrome using the local MSPB RMUAMT app for the Red or Orange Class of Service
; only if it is not already installed. Refer to the Class of Service operation guide
; for more information about configuring Class of Service controls.
RS:RMUAMT -A:INSTALL --P:CHROME=Orange,Red;-GCH

[ALL_EXCLUDE]
[ALL-WKSTNS_EXCLUDE]
WIN-Office 365 - Install - 2019Office Pro Plus=cust1,cust2,cust3
; cust1 installs from their local server, cust2 uses OpenOffice, cust3 excludes Teams
[ALL-SERVERS_EXCLUDE]
; These procedures will be executed on all platforms for the specified customer
[custid-ALL]
; These procedures will be executed on all workstations for the specified customer
[custid-wkstns]
BlueCoat Security=All
; These procedures will be executed on all workstations for the specified customer/site
[custid-wkstns.site-id]
; These procedures will be executed on all servers for the specified customer
[custid-servers]
```

Log File

RMUOBA_*#-DAY*.LOG

RMUARN - Agent Rename Utility

Rename an agent and optionally reboot

Renames the computer and automatically reboots to activate the renaming operation. The reboot operation can be suppressed with an optional argument, but the rename operation will not take effect until the computer is restarted.

Platforms:

Runs on any RMM platform.

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--N:<name>	REQUIRED - Define the new computer name.
--A	REQUIRED - use the agent name to set the computer name. This is valid only when the RMM platform name can be set independently of the computer name.

ONE of the above two parameters are required. If both are used, the first one specified will be used.

--R	OPTIONAL - Suppress the reboot - default is to reboot immediately.
--D	OPTIONAL - Enable debug mode - log but perform no actions.

If the specified name already matches the current computer name, then no action is performed.

Config File

No configuration file is used.

Log File

RMUARN.LOG

RMUAAC - Verify RMM Platform API Access

Connects to the RMM platform's API interface to verify access

This tool runs from the agent computer and attempts to connect to the RMM Platform's API interface. The endpoints often use the API interface to update audit data, obtain configuration data, or initiate automation procedures. The tools utilize either User/Password (using a calculated password) or Token access. This tool prefers Token-based authentication and will test that first, then fall-back to user/password authentication.

Use this tool to determine if the API URL is being blocked or if the User/Password or security tokens are being properly used.

The RMM Suite will never cache the password or store the access token in plain text. When User/Password authentication is used, the password is generated dynamically. Access tokens are stored using a secure cipher and deciphered when needed.

Platforms:

Runs on any RMM platform.

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

None

Config File

No config file is used.

Log File

RMUAAC.LOG

RMUAMT - Third-Party App Management Tool

Installs, Updates, or Removes third-party applications

Supports 3 methods of application management:

- NINITE - uses Ninite Pro (Classic) to Install, Update, or Remove applications.
- CHOCO - Uses Chocolaty to Install, Update, or Remove applications.
- DIRECT - uses internal logic to download and install or remove applications. Updating via the DIRECT method is not supported. This is generally used for custom LOB applications.

The NINITE and CHOCO methods require a combined additional license fee to use these commercial products. This license is available through MSP Builder. Contact MSP Builder sales for pricing. App Management will exit silently without taking any action if a license is not found.

The configuration file defines the primary method that will be used to execute the commands, along with the arguments needed to perform the requested action. If the primary method requires a license and the customer is not currently licensed, the action will terminate with a FAIL status. Select applications may support a DIRECT method in addition to the NINITE or CHOCO method. These configurations will fall back to DIRECT operation when licenses are unavailable and direct install methods are defined.

The choice of NINITE, CHOCO, or DIRECT is fixed - the optimal method will be used whenever possible.

Note that NINITE and CHOCO methods will install or update to the latest available version, while the DIRECT method will install a specific current version that may not be the latest available.

Platforms:

Runs on any RMM platform.

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--P:<PRODID>	REQUIRED - Defines the product to process. This is a package file obtained from and managed by MSP Builder. This consists of a 6-8 character product code. A special form of PRODID is “LIST:*name*”, where “*name*” identifies a Cloud Script Variable that contains a comma-delimited list of Product IDs.
--A:<ACTION>	REQUIRED - Defines the action to perform - must be one of INSTALL, REMOVE, or UPDATE.
--T	Suppress terminating running apps during an UPDATE cycle. Normally, applications that are running will be terminated prior to updating.
--F	Force-updates all apps even if they are defined in an Exclude list.
--U	Initiates an Update All action - the equivalent of --P:ALLAPS --A:Update.
--LIST	List all currently supported package codes and application names.

Config Files

RMUAMT.TXT - Defines the core configuration arguments for the select method (Ninite, Chocolaty, or Direct) and the application codes supported by each method. The configuration file is maintained by MSP Builder and is not customizable. This file is not physically deployed to endpoints but loaded directly into the application from the cloud server to enhance security. See the list below for available **Product Codes**.

RMUAMT.INI - Defines two operational controls:

PermitAppManagement	Disables Application Updates on the target computer. INSTALL and REMOVE actions are still permitted.
SuppressAMTerminate	Prevents terminating running applications during updates.

Cloud Script Variables

The App Management tool supports the use of Cloud Script Variables to further configure the operation.

AMExclude	This is a specific variable used to define a comma-delimited list of package codes that should be excluded from updating. Any package code defined in this list will not be able to be installed, updated, or removed. This applies to specific action tasks (install CHROME when CHROME is excluded) as well as any LISTs or Update All processes. The Exclusion can be overridden by specifying the --F parameter. Normally, this application is invoked weekly to update all installed applications. This option can be set globally, by customer, site or department, or specific device to suppress updating selected applications.
--P:**LIST:<CSVName>**	Any CSV parameter can be used to define a comma-delimited list of package codes to perform an action upon. For example, **`AMBASIC=ACROBT,CHROME,SVNZIP`** can define a list of commonly deployed applications that can quickly be installed and/or updated. We recommend that the CSV name you define begin with "AM" to indicate it is related to App Management. The actual name is unimportant and should convey the purpose of the grouping. There are no limits on the number of application groups that can be created, although each group requires a uniquely named CSV. By using "--P:LIST:CSVname" a user of the RMM Suite can create custom deployment sets to rapidly deploy collections of applications. Deployment can be done by RMM Script or through OBA/CSM

Log File

RMUAMT_*#-DAY*.log

The results of the actions are recorded to specific files. These files are combined (for LIST and Update All processes) into the CmdResult.log file. This is then appended to the primary RMUAMT log file, which also contains information related to the general operation of the application.

Status File

AMStatus.INI

This file is located in the main MSPB folder and reports the status of all installed apps. Each supported app has an entry similar to:

```
[Chrome]
CurrentVersion=121.0.6167.185
AvailableVersion=None
AppStatus=CURRENT | DOWNLEVEL
Update=None | VERSION_NUMBER
LastCheckDate=2024/02/15
LastUpdateDate=2024/02/15 | NEVER
LastAction=PASS | FAIL
```

While most entries are self-explanatory, alternate values are shown. When an application is “DOWNLEVEL” the Update will report the available version to install. The LastUpdateDate will report “NEVER” if it has never been updated by App Management. This data is used to update the MSPB cloud reporting portal but is available locally for custom logic.

Product Codes

The product codes below represent the various applications that are supported by App Management at the time of publication. These will automatically select the appropriate method for performing the requested action - either Ninite, Choco, or SWD, and execution is dependent on an appropriate license entitlement.

CODE	NAME	CODE	NAME
DOTNET	.NET	CCCP	CCCP
.NET_4	.NET 4	CCLEAN	CCleaner
.NET_3.5	.NET 3.5	CDBurnerXP	CDBurnerXP
.NET_4.5	.NET 4.5	CHROME	Chrome
.NET_4.5.1	.NET 4.5.1	CTXRCV	Citrix Receiver
.NET_4.5.2	.NET 4.5.2	CTXWSP	Citrix Workspace
.NET_4.6	.NET 4.6	CLSTRT	Classic Start
.NET_4.6.1	.NET 4.6.1	CUTPDF	CutePDF
.NET_4.6.2	.NET 4.6.2	DIGSBY	Digsby
.NET_4.7	.NET 4.7	DSCORD	Discord
.NET_4.7.1	.NET 4.7.1	DRPBOX	Dropbox
.NET_4.7.2	.NET 4.7.2	ECLPSE	Eclipse
.NET_4.8	.NET 4.8	EDGE	Edge
.NETDR-5	.NET Desktop Runtime 5	EMULE	eMule
.NETDR-6	.NET Desktop Runtime 6	MSESSN	Essentials
.NETDR-7	.NET Desktop Runtime 7	EVRNOT	Evernote
.NETDR-8	.NET Desktop Runtime 8	EVRYTG	Everything
.NETDR-5x64	.NET Desktop Runtime x64 5	FSTSTN	FastStone
.NETDR-6x64	.NET Desktop Runtime x64 6	FLZLLA	FileZilla
.NETDR-7x64	.NET Desktop Runtime x64 7	FIRFOX	Firefox
.NETDR-8x64	.NET Desktop Runtime x64 8	FFXESR	Firefox ESR
SVNZIP	7-Zip	FFXESR10	Firefox ESR 10
ACROBT	Acrobat	FFXESR17	Firefox ESR 17
ARDRDC	Acrobat Reader DC x64	FFXESR24	Firefox ESR 24
ADAWRE	Ad-Aware	FFXESR31	Firefox ESR 31
AIM	AIM	FFXESR38	Firefox ESR 38
AIMP	AIMP	FFXESR45	Firefox ESR 45
ADBAIR	Air	FFXESR52	Firefox ESR 52
AUDCTY	Audacity	FFXESR60	Firefox ESR 60
AUSLGC	Auslogics	FFXESR68	Firefox ESR 68
AVAST	Avast	FFXESR78	Firefox ESR 78
AVGAV	AVG	FFXESR91	Firefox ESR 91
AVIRA	Avira	FFXESR102	Firefox ESR 102
BITWRD	Bitwarden	FFXESR115	Firefox ESR 115
BTRRNT	BitTorrent Sync	FLASH	Flash
BLENDR	Blender		

CODE	NAME	CODE	NAME
FLSHIE	Flash (IE)	JAVART-7x64	Java x64 7
FLSHPP	Flash (PPAPI)	JAVART-8x64	Java x64 8
FOOBAR	foobar2000	JAVART-9x64	Java x64 9
FXTRDR	Foxit Reader	JAVART-10x64	Java x64 10
GIMPIP	GIMP	JDK	JDK
GLARY	Glary	JDK-6	JDK 6
GOM	GOM	JDK-7	JDK 7
GGLBAK	Google Backup and Sync	JDK-8	JDK 8
GGLDRV	Google Drive	JDKAO	JDK (AdoptOpenJDK)
GDRVFS	Google Drive File Stream	JDKAO-8	JDK (AdoptOpenJDK) 8
GDRVDT	Google Drive for Desktop	JDKAOx64	JDK (AdoptOpenJDK) x64
GEARTH	Google Earth	JDKAO-8x64	JDK (AdoptOpenJDK) x64 8
GTALK	Google Talk	JDKAO-11x64	JDK (AdoptOpenJDK) x64 11
GOTMTG	GoToMeeting	JDKAO-17x64	JDK (AdoptOpenJDK) x64 17
GRNSHT	Greenshot	JDKAO-21x64	JDK (AdoptOpenJDK) x64 21
HNDBRK	HandBrake	JDKAC	JDK (Amazon Corretto)
HULU	Hulu	JDKAC-8	JDK (Amazon Corretto) 8
IMGBRN	ImgBurn	JDKACx64	JDK (Amazon Corretto) x64
INFRCD	InfraRecorder	JDKAC-8x64	JDK (Amazon Corretto) x64 8
INKSKP	Inkscape	JDKAC-11x64	JDK (Amazon Corretto) x64 11
IRFNVW	IrfanView	JDKAC-17x64	JDK (Amazon Corretto) x64 17
ITUNES	iTunes	JDKAC-21x64	JDK (Amazon Corretto) x64 21
JAVART	Java	JDKx64	JDK x64
JAVART-6	Java 6	JDK-7x64	JDK x64 7
JAVART-7	Java 7	JDK-8x64	JDK x64 8
JAVART-8	Java 8	JDK-11x64	JDK x64 11
JRTAO	Java (AdoptOpenJDK)	KLTCDC	K-Lite Codecs
JRTAO-8	Java (AdoptOpenJDK) 8	KLTCDCx64	K-Lite Codecs x64
JRTAOx64	Java (AdoptOpenJDK) x64	KEEPSC	KeePass
JRTAO-8x64	Java (AdoptOpenJDK) x64 8	KEEPS2	KeePass 2
JRTAO-11x64	Java (AdoptOpenJDK) x64 11	KMPLYR	KMPlayer
JRTAO-17x64	Java (AdoptOpenJDK) x64 17	KRITA	Krita
JDRTO-21x64	Java (AdoptOpenJDK) x64 21	LNCHY	Launchy
JRTORC-8	Java (Oracle) 8	LIBREO	LibreOffice
JRTORC-8x64	Java (Oracle) x64 8	LGMEIN	LogMeIn
JAVARTx64	Java x64	MWBYTE	Malwarebytes
JAVART-6x64	Java x64 6	MMONKY	MediaMonkey

CODE	NAME	CODE	NAME
MSMSGR	Messenger	SHKWAV	Shockwave
MOZYBK	Mozy	SLVRLT	Silverlight
MSCBEE	MusicBee	SKYDRV	SkyDrive
NTPDPP	Notepad++	SKYPE	Skype
NVDA	NVDA	SNGBRD	Songbird
ONEDRV	OneDrive	SPOTFY	Spotify
ONPNSHL	Open-Shell	SPYBOT	Spybot
OPNOFC	OpenOffice	SPYBOT-2	Spybot 2
OPERA	Opera	SPYBOT-1	Spybot 1
OPERAC	Opera Chromium	STEAM	Steam
OPERA-1	Opera 12	SGSYNC	SugarSync
PNTDNT	Paint.NET	SMTPDF	SumatraPDF
PDFCRT	PDFCreator	SUPRAV	Super
PEAZIP	PeaZip	TMVWER	TeamViewer
PICASA	Picasa	TMVWER-12	TeamViewer 12
PIDGIN	Pidgin	TMVWER-13	TeamViewer 13
PUTTY	PuTTY	TMVWER-14	TeamViewer 14
PYTHON	Python	TMVWER-15	TeamViewer 15
PYTHN-3	Python 3	TRACPY	TeraCopy
PYTHN-3x64	Python x64 3	THDBRD	Thunderbird
QBTRRNT	qBittorrent	TBDESR	Thunderbird ESR
QKTIME	QuickTime	TRILLN	Trillian
READER	Reader	TRUCPT	TrueCrypt
READER-9	Reader 9	TWDECK	TweetDeck
READER-10	Reader 10	UTRRNT	uTorrent
READER-11	Reader 11	VSCODE	Visual Studio Code
READER-DC	Reader DC	WINVLC	VLC
ARDRCL	Reader Classic	WEBEXC	WebEx
ARDRCL-15	Reader Classic 2015	WINAMP	Winamp
ARDRCL-17	Reader Classic 2017	WNDRST	WinDirStat
ARDRCL-20	Reader Classic 2020	WMERGE	WinMerge
RELVNC	RealVNC	WINRAR	WinRAR
RVNCSV	RealVNC Server	WINSCP	WinSCP
RVNCVW	RealVNC Viewer	WIZTRE	WizTree
REVOUN	Revo	XNVIEW	XnView
SAFARI	Safari	YMSNGR	Yahoo!
SHAREX	ShareX	ZOOMDT	Zoom

RMUCET - Command Execution Tool

This is an RMM Suite application that downloads a command or script along with optional support files and then executes the command/script. Components are downloaded into the RMM Suite tools folder to eliminate the need to whitelist other installation and execution folders and minimize the security footprint. Deploying these tools and files from the MSP Builder cloud allows them to be updated and maintained without hosting content directly on the RMM platform.

- Any executable, MSI, batch, VB, or PowerShell script can be executed - no limit to script complexity or length.
- Any number of additional files can be downloaded, either from the MSPB cloud or via an HTTPS URL (MSP or vendor website). ZIP files are automatically unzipped.
- Supports macros to identify MSPB Client ID, MSP customer ID to provide granular poptions for configuration file downloads. These files can be tagged as "OPTIONAL".
- MSPB Cloud Script Variables can be defined as either required or optional parameters and perform Macro replacements in the command / script. A missing required CSV will terminate the execution of the process.

Any errors will terminate the execution and report status to the calling process.

MSPB can provide custom packages, including hosting of files, at no additional cost to support any custom application deployment, configuration, or monitoring requirement.

Platforms:

Runs on any RMM platform.

Usage:

Invoked from the RMM platform, Daily Maintenance, Onboard Automation, or locally via the **RSRUN.BAT** script.

Arguments

--C:<PKGID>	Defines the Package ID that should be downloaded and executed. Scripts may be BAT or PowerShell based and may include additional executable and configuration files where needed. Files are downloaded into the MSPB\Work\<PKGID> folder, executed, and then removed when the task has been completed.
"NAME=value"	Represents a Macro Name and the corresponding value to be defined in a command or script. MUST be enclosed in quotes as shown.
Argument…	Zero or more general arguments that can be passed to a command or script, either as positional arguments or in a specific position via a macro.

Config File

No local configuration file is used.

Log File

RMUCET_<PKGID>.LOG

The script / application that is executed may generate its own log (cmdresult.log). That log is appended to the application log before the folder is removed.

Additional Reference

Refer to the RMM Suite Operations Guide - 18 - Command Exec Tool for details on configuration and operation of this tool.

RMUDFS - Disk Free Space Tool

This application requires two arguments - the Disk and the Size in MB. It will determine the amount of free space on the requested disk, compare it to the size requested, and return “PASS” if the disk has sufficient free space available and “FAIL” if it does not. This tool is designed to be called from RMM platform scripts to decide if enough space is available to perform the required actions of the script, without needing any math functions in the script.

Platforms:

Runs on any RMM platform

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

D:	The disk drive to check. Must be the first argument
###	The size, in megabytes, that is required for the task to succeed. The actual free space is compared to this value.

Config File

No configuration file is used.

Log File

No log is created.

RMUDVR - Generate a Disk Volume Report

Reports the number of files and size, by directory, for a given path. Can show the full detail of all files found and can track utilization totals as well as totals by age. This creates a CSV file in C:\Temp\DiskVol.csv and is uploaded upon completion where supported by the RMM platform.

Platforms:

Runs on any RMM platform

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--P:path	The full pathspec to examine.
--F:file	Specify an alternate output file path and name.
--D:yyyy/mm/yy	Files older than the specified date will be reported separately.
--A	Report based on last access time instead of last modified time when the --D argument is used.
--R	Summarize counts at the first subfolder.
--FD	Display *Full Detail* - list every file

Config File

No configuration file is used.

Log File

RMUDVR.LOG	Logs operation. Data is recorded in a separate CSV file.

RMUELC - Execute a Local Command (deprecated)

Deprecated *- replaced by RMUCET with the same capabilities.*

This is an internal-use application used by the RMM Suite to execute commands that exist on the endpoint. The actual command and arguments are delivered from the MSP Builder cloud, allowing the commands to be updated without delay or need to modify RMM platform scripts.

Platforms:

Runs on any RMM platform

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--C:<code>	Defines the code associated with the command. The code is sent to the MSP Builder cloud server and a secure command string is returned for processing. Commands can support Cloud Script Variables for automatic injection of client-specific arguments.

Config File

No configuration file is used.

Log File

RMUELC.LOG

The output of the command is logged, and that log is appended to the primary log file. Where supported, the entire log file is collected and stored on the RMM platform.

RMUCSV - Get Cloud Script Variable

Customers can define Cloud Script Variables (CSVs) to define data that can be injected into RMM Suite applications or custom scripts. This allows a single script to obtain data such as license keys, credentials, and other settings on a customer, location, or agent-specific level.

This tool takes a single argument - the name of a Cloud Script Variable. It then obtains the data associated with that variable and outputs it, where it can be captured by scripts or RMM platform processes.

If a requested variable is not defined, the application returns "-UNDEFINED-". If the data has been ciphered, it returns "-SECURE-". The application cannot obtain, decipher, and output data that has been ciphered to ensure the security of that data. Only RMM Suite applications have the ability to decipher and use the ciphered data values.

It an error is encountered trying to obtain the Cloud Script Variables, the application outputs "FAIL - ", followed by the exact error message. Any custom scripts that leverage CSVs should test for the FAIL, SECURE, and UNDEFINED values and handle them appropriately.

Platforms:

Runs on any RMM platform

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

CSV_Name	The name of the Cloud Managed Variable to return the data for. Names can be determined via the Configuration Management interface, and custom values can be created and defined.

Config File

No customizable configuration file is used. Obtains data from the local RMUCSV.INI file.

Log File

No log file is generated.

RMUGES - Get and Execute Script (deprecated)

***Deprecated** - replaced by **RMUCET** with the same capabilities.*

This is an RMM Suite Internal-Use application that downloads a script along with optional support files and then executes the script. Components are downloaded into the RMM Suite tools folder to eliminate the need to whitelist other installation and execution folders and minimize the security footprint. Deploying these tools and files from the MSP Builder cloud allows them to be updated and maintained without hosting content directly on the RMM platform.

Platforms:

Runs on any RMM platform

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--C:<code>	Defines the Script ID that should be downloaded and executed. Scripts may be BAT or PowerShell based and may include additional executable and configuration files where needed. Files are downloaded into the MSPB\Work\<code> folder, executed, and then removed when the task has been completed.

Config File

No configuration file is used.

Log File

RMUGES.LOG

The script / application that is executed may generate its own log. That log is appended to the application log and - where supported - the log is uploaded to the RMM platform as a record of the execution. Not all RMM platforms support file uploading.

RMUGLZ - Get Diagnostic Data as Zip File

This application should be used whenever opening a support ticket with MSP Builder. It will collect the RMM Suite log files, configuration files, and dump the registry configuration data. This information provides the MSP Builder support team with everything they need to solve most issues.

Platforms:

Runs on any RMM platform

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

No arguments are used.

Config File

No configuration file is used.

Log File

No log file is generated.

A **RSConfigData_*identity*.zip** file is generated in the C:\Temp folder of the endpoint. Where supported, the file is uploaded to the RMM platform when the data collection is completed. **This file is automatically attached to an email sent to MSP Builder support.**

RMUINI - Manipulate INI File Data

This is a customer-use tool to provide an interface between RMM platform scripting and INI format configuration files located on the endpoint. The too can target a file, section, and value and perform a read, write, or update operation. The command requires multiple parameters to function, and the use of parameters may vary depending on the operation being performed.

Platforms:

Runs on any RMM platform

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--A:*value*	Defines the action - Read, Write, or Update. Update is used with the --I option to merge the contents of one INI file (specified by --I:file) into the target file specified by --F:file.
--F:*file*	Defines the full file and path of the INI file where the operation will occur.
--S:*value*	Specifies the section in the INI file to target.
--V:*value*	Specifies the value within the section to read or update. If the value is the empty string (“”), the entire section will be removed from the INI file.
--D:*value*	Valid only with a Write action, it defines the value to write to the section:value. If the value is the empty string (“”), the value will be removed from the INI file.
--I:*file*	Defines the full file and path of an INI file to merge into the file defined by --F:file. When the Action is “update”, only the --F and --I arguments are used.

Config File

No configuration file is used.

Log File

No log file is generated. Values read are returned via STDOUT, where they can be captured by other scripts. Failures are also written to STDOUT, starting with “FAIL-<ID> - Error Message”. The script that invokes this application should trap for these FAIL messages and handle them appropriately. Actions (such as Write or Update) that do not return a specific value will return “OK” if successful.

RMUISP - Internet Speed Test

This tool can be used stand-alone to return an immediate assessment of average Internet speed, create a detailed report of multiple speed tests with min/max and average speeds, and even be invoked through Daily Maintenance to write data to the Audit cache file, where the Daily Audit tool can update a custom field with the speed results.

Caution: This application will place a 2-15% CPU load on the endpoint for 2-3 minutes.

Platforms:

Runs on any RMM platform

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--A	Update the audit cache with Date, Min, Max, and Avg speed values.
--C:#	Specify the number of downloads per file size. The allowed range is 1-4, and the default counts are 4, 4, 2, and 2 (for 4MB, 16MB, 64MB, and 256MB).
--S:#	Specifies the start size - default is 1 in the range of 0 to 3. The 4MB file size is not normally used due to the extra processing overhead, which tends to artificially reduce the throughput. 0 = 4MB, 1=16MB, 2=64MB, and 3=256MB
--E:#	Specifies the end size in the range of 1-3 - default is 3. Invalid values for Start or End will use the default size.
--F	Fast Mode. Normally, the tool will ensure that the test files are cached at the web server to eliminate all File I/O overhead. Fast Mode eliminates this pre-loading, and can be used when repeated tests are performed.

Config File

No configuration file is used.

Log File

RMUISP.LOG

RMUSCM - Manage MS Security Center

Display or De-register AV Products in MS Security Center

Without arguments, lists the product names and GUIDs for every AV product registered in Microsoft Security Center. When used with an argument and specifying either the product name or GUID, that product / GUID will be de-registered from Security Center.

This task may be required when uninstalling a security product does not properly deregister from Security Center. This may result in RMM Suite detecting "ghost" products and reporting multiple products or products not updating properly.

Platforms:

Runs on any RMM platform

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--V	Visual (screen) Logging - must be a live console connection. This may not display on certain remote management tools.
--X:<guid>	Remove the specified GUID from Security Center. Use this to remove a specific GUID from Security Center.
--R:<name>	Remove the specified NAME product(s) from Security Center. Note that this will remove ALL products from Security Center that match the defined name!

Config File

No configuration file is used.

Log File

RMUSCM.LOG

RMUODJ - Offline Domain Join Utility

Performs a fully-unattended offline domain join operation

This tool works in conjunction with an RMM Platform script on supported platforms. It starts by locating a Domain Controller agent in the RMM platform associated with the active customer. It then invokes an RMM Script on the domain controller, passing key information from the agent. The Domain Controller creates the domain object and returns a binary data file that is returned to the agent, which completes the domain join process using that data file.

Platforms:

Requires RMM Platform Features: Agent Search with Filter, Execute Command, Get File, Download File

Kaseya VSA

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--D	Enabled DEBUG Mode - commands are logged and not executed.

Config File

No configuration file is used,

Log File

RMUODJ.LOG

RMUSPM - System Policy Manager

Accepts an argument and updates the Policy Control value, returning the result

The RMM Suite uses a combination of values to enable and disable automation. The Policy Control field is an agent-specific value that can contain Tags that are used to disable specific automation features. These tags need to be added or removed.

This tool accepts an argument representing a tag and an action. The action performs an add or remove of the specified tag from the Policy Control data field. This data is maintained in the registry of the local computer for use by local applications and is synchronized with custom fields on the RMM platform to control RMM-based automation.

Platforms:

Runs on any RMM platform

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

+<tag>	Add the identity to the PolicyMgmt key. Note the leading "+" character.
-<tag>	Remove the identity from the PolicyMgmt key. Note the leading "-" character.
-CLEAR-	Clears the PolicyMgmt key. Note the leading and trailing "-" characters.

Config File

No configuration file is used

Log File

RMUSPM_*#-DAY*.LOG

RMUAIB - Agent Initialization and Branding

Performs RMM-Specific configuration actions

When a new agent is installed, this utility will perform specific configuration actions according to a configuration file. These tasks are often implemented by the MSP or IT department to "personalize" the RMM platform software. This tool performs the following tasks:

- Adds the MSPB\Bin folder to the path so tools can be run from a command prompt.
- Defines the RSWorkDir environment variable - other tools can reference this to find configuration files or locate additional tools in the Bin subfolder.
- Disables sleep/hibernate for AC Power in active power config.
- Sets agent/platform-specific branding options, such as:
    - Renaming the RMM service to include the company name
    - Remove the Agent Uninstaller from the Add/Remove Programs menu.
    - Remove the RMM Agent from the start menu.

Platforms:

Runs on any RMM platform, although specific tasks may differ slightly between RMM platforms.

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--f	Forcibly reprocess the initialization. Most initialization tasks are locked to a one-time only execution.

Config File

RMUAIB.INI - Two sections control agent environment and RMM software settings.

The INIT section defines two parameters that change how the computer functions:

- SetPowerOptions — Disables setting the power configuration if false (default is Yes)
- DisableSCA — Disable non-critical or All Security Center pop-up alerts (default is No)
  Options are No, Yes, or ALL

The following values are located in the BRANDING section and are RMM Platform-Dependent

- DelStartMenu — Remove the agent from the Start Menu if true (default is Yes)
- UpdateRun — Update the Run Once if true (default is No)
  Some RMM platforms insert a Run Once action that this will disable.
- RemUninstall — Remove the Agent Uninstaller if true (default is Yes)
- UpdateSvcId — Rename the Service Descriptive Name if true (default is Yes)
- HideWorking — Hide the Working folder if true (not recommended/default is No)

Log File

RMUAIB.LOG

RMURDT - Run Daily Tasks

Run all Daily Init and Automation Tasks

This tool performs all RMM Suite endpoint initialization, content management, and automation management without additional communication from the RMM Platform. This tool runs in two phases - PREP and EXEC.

PREP Phase

The tools are downloaded from the MSP Builder distribution server via the RMM Platform. The RMM Platform script then initiates the app, which requests client configuration and licensing data from the MSP Builder cloud services. All required folders are verified and created as necessary, the configuration data is written to the RMM key in the registry, and if this is the first time that the process has run, it performs the agent initialization and branding tasks. The application is then re-executed to start the EXEC phase.

EXEC Phase

In the EXEC phase, the application performs the following actions:

- Removes any Temp files from the PREP phase
- Downloads and/or updates all RMM Suite applications.
- Identifies and downloads the MSP or Org-Specific branding files for the User Interface.
- Removes and downloads all application configuration files.
- Runs a Quick Audit, updating the local cache data with select data that can be used for Zero-Day remediation tasks.
- Runs the Daily Maintenance Sequencer to run all appropriate maintenance tasks
- Runs the Smart Monitor Sequencer to run all appropriate Smart Monitors
- Runs the Full Audit to collect all local asset data, then upload that data to the RMM platform. Select configuration data that rarely changes is collected monthly, while most audit data is collected and updated daily.

The RMURDT application is suited for secure environments and can be configured to obtain endpoint configuration data from an MSP or IT Department hosted facility. Only the customer/license data is obtained from MSP Builder - all other files are downloaded from the MSP's designated deployment server. Contact MSP Builder sales team for more information on self-hosting capabilities.

Platforms:

Runs on any RMM platform

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--Audit	OPTIONAL: Update apps and configs, then run full audit
--A	OPTIONAL: Update apps and exit
--B	OPTIONAL: Force update BMS scripts only
--C:CustID	REQUIRED: Define the Customer ID in MSP_Identity if set (pre-run stage only)
--C	OPTIONAL: Update apps, download configs, then exit (also DAILY blocker)
--D	OPTIONAL: DEBUG Mode - suppress select actions, provide additional logging
--F	OPTIONAL: Force download all objects
--I	OPTIONAL: Perform platform initialization (Agent Branding, Onboard Automation, Full Audit)
--L	OPTIONAL: Install Locally (Autonomous Operation Mode)

--MD:metadata	Pass a set of RMM-specific name=value pairs. This allows passing of RMM data, custom fields, and other data to configure operation.
--RT	Create or recreate a new, randomly scheduled task for autonomous daily operation. Generates a random time within a 1-hour (servers) or 4-hour (workstations) window.
--RX	Recreate the scheduled task using the time defined in the DTSchedRun registry key. Used to either recreate a deleted task at the original time or to modify the registry time and update the scheduled task to use the new time. CAUTION: The start time must be within the 07:00 to 16:30 time range for automation to be effective.
--U:url	OPTIONAL: Download URL Override - USE WITH CAUTION
--X	Used only by the AOM scheduled task to initiate the tasks from the C:\Temp\mbrs folder.

Config File

No configuration file is used, however, this tool downloads, deciphers, and extracts the configuration files used by all other RMM Suite tools.

Log Files

PREP Phase: RMURDT_PREP.LOG
EXEC Phase: RMURDT_*#-DAY*.LOG

AOM Mode only: RMURDT_LOCAL_*#-DAY*.LOG

RMUSNAP - System Snapshot Tool

This application is often run from Daily Maintenance to generate a snapshot of services, applications, and processes that are running on the endpoint. This can provide a baseline of what's "normal" to aid in determining unusual issues on the endpoint. Snapshots can be run interactively, and these use a different naming format to allow comparing the normal and the current environmental conditions.

Platforms:

Runs on any RMM platform

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

<none>	Creates a normal snapshot named for the current day.
--M	Creates or updates a "master" file to establish baselines of what should be running under normal conditions. This is usually created immediately after a reboot of the endpoint, before any user applications start.
--I	Creates an Immediate file that can be used to compare the current system activity with the master or daily data.

Config File

No configuration file is used

Log File

RMUSNAP_*#-DAY*.LOG

Each collection creates a CSV format file in the MSPB\Logs subfolder. These can easily be imported into Access, SQL, or Excel for reporting or comparisons.

RMUSUL - System User Logoff Tool

This tool is used to forcibly initiate the logoff of a specific user or all users on a target computer. This is often used on RDS hosts prior to rebooting to ensure that users are logged off before the reboot. An optional message will be displayed to provide a generic “rebooting in 2 minutes” warning. The logoff action occurs immediately when the message is acknowledged.

Platforms:

Runs on any RMM platform

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--U:*name*	REQUIRED - Define the specific user to log off. This can be specified multiple times to log off multiple specific users. The name “ALL” will cause all users to be determined and individual logoff actions initiated on each user.
--R	Displays a “Rebooting in 2 minutes - save your work and click OK.” message to all logged-in users. Only valid with the “ALL” user argument. DOES NOT INITIATE A REBOOT.

Config File

No configuration file is used

Log File

RMUSUL_*#-DAY*.LOG

RMUUAM - Manage Local User Accounts

Create, Delete, or Modify local user accounts and group membership

Used to create an account, add it to the local administrators group (can be suppressed), or update the password of an existing account. It can also delete accounts or add the account to or remove the account from a specified local security group. The tool does not distinguish between create and update, it automatically will determine if the account should be created or only have it's password updated.

When the Agent Init process is executed, this application is invoked twice with the following arguments:

```
RMUUAM.BMS --A --U:CSV_RAUserID --P:CSV_RAPassword --N:CSV_RAUserName --HV
RMUUAM.BMS --A --U:CSV_CAUserID --P:CSV_CAPassword --N:CSV_CAUserName --HV
```

This will automatically create (or update) the MSP's local admin account based on the "RA" values, and the Customer's local admin account (where needed) based on the "CA" values. If either the UserID or Password are blank, the account will not be created or updated. This is not an error condition. The UserName value is optional and will be set if defined, otherwise it will be ignored.

Platforms:

Runs on any RMM platform.

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script. This is also executed as noted above when the Daily Tasks application runs for the first time or is forced to run in "Init" mode.

Arguments

--U:str	Required - the user account name to manage. If the value is prefixed with "CSV_", the data following this prefix should identify a CSV value name, and that value will be used as the user ID.
--P:str	Required - the user account password to set. If the value is prefixed with "CSV_", the data following this prefix should identify a CSV value name, and that value will be used as the password. If the password was ciphered using the standard MSP Builder tools, the key will be generated and the password deciphered before use.
--N:str	Optional - the user account full name, if creating the account. If the value is prefixed with "CSV_", the data following this prefix should identify a CSV value name, and that value will be used as the user account name.
--A	Optional - Create an admin account, default is a regular account. This will add the account to the local Administrators group.
--H	Optional - Hides the account
--HV	Optional - Hides the account if the corresponding CSV is set to TRUE (CA/RAUserID-HIDE=Y) --HV MUST be specified after the --U parameter to properly associate the action with the specific account name!
--DC	Optional - Permit operation on Domain Controllers (normally blocked)
--X	Optional - Delete the specified account
--G:name;action	Optional - Add/Remove the user from the specified group. Action should be either Add or Remove
--C	Optional - Password supplied is CLEAR-TEXT! When RMM Suite tools invoke this application, they pass the password as a ciphered string using a 16K-bit cipher key. This option allows an RMM Platform script to prompt the user for credentials and pass them to our application in plain-text.

Special Notes for --HV Option

When using the --HV option, the --HV must follow the --U: argument so it can map to the specific UserID defined by the --U argument. We recommend that --HV be the last argument specified whenever possible to avoid any confusion.

The --HV argument uses the parameter associated with --U:<userid>. Since the UserID can either be an actual User ID or a *reference* to a UserID CSV value, it performs two distinct matching actions according to the following rules:

--U:MyAdmin --HV	This looks for a CSV called "*MyAdmin*-HIDE=Y" and will hide the account if found. In this example, the account being created is "MyAdmin".
--U:CSV_RAUserID --HV	This method is a bit more complex in that the "CSV_" prefix tells the app to look up the actual UserID in the CSV list, and that the lookup ID is "RAUserID". The CSV list likely contains an entry similar to "RAUserID=itadmin". This defines "itadmin" as the actual UserID. The application will attempt to locate either the lookup reference or the lookup value. Specifically, if either" RAUserID-HIDE=Y" or "itadmin-HIDE=Y" are defined, the account will be hidden.

The second method above is used by the New Agent Init logic to create a local admin account when the RAUserID and RAPassword values are defined for a customer. The hide option can be specified as a global (all customers) or a customer-specific override as appropriate.

Config File

No configuration file is used. Cloud Script Variables are read from the local RMUCSV.INI file.

Log File

RMUUAM.LOG

RMUWPU - Windows Platform Upgrade

Performs an "Any to Latest" upgrade of the targeted Windows computer

Upgrades Windows 10 or 11 to the latest Build or can upgrade Windows 10 to the latest Build of Windows 11. Any Build can be targeted without any specific prerequisites.

This tool will first perform specific checks to ensure that the endpoint is ready to perform an upgrade. It extracts the current Version and Build ID and issues a request to Microsoft to download the correct update package to upgrade to the latest available version of Windows. The "latest" version is controlled by MSP Builder and is updated approximately 2-3 weeks after Microsoft releases the update.

The tool performs the following tests and will exit with an error code and message on the first failure it detects:

- "FAIL-CONFIG" Unable to contact dist.mspbuilder.com for config data.
- "FAIL-AUTH" Unable to perform software license validation.
- "FAIL-API" Unable to load the RMM API module.
- "FAIL-LICENSE" Windows License is not Activated.
- "FAIL- SPACE " Insufficient disk space available (< 32GB).
- "FAIL-VERSION" Installed O/S Version below minimum supported Version.
- "FAIL-W11HWSCRIPT" Unable to download Hardware Readiness data.
- "FAIL-W11HWCHECK" Failed the Windows 11 Hardware Readiness checks.
- "FAIL-REBOOT" Pending Reboot status is active.
- "FAIL-DOWNLOAD#" Failed primary or backup download operation.

If the computer is already at the highest Version:Build, the applications exits with a success message "OK-Current".

Bitlocker Support

If Bitlocker is active and protection is enabled, protection will be disabled before starting the upgrade process. The Bitlocker change status will be recorded, and protection will be re-enabled when this tool is run in verification mode after the upgrade completes.

NOTE: This tool is certified for use for upgrading any Windows 1x Build to the latest Build on the same version or upgrading from Windows 10 to Windows 11 latest Build. It will not be supported for upgrading from Windows 7/8x to Windows 1x or upgrading server operating systems.

Platforms:

Runs on any RMM platform.

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script. There are usually 3 RMM Scripts:

- Verify - verify the endpoint meets all upgrade requirements.
- Upgrade - initiate the upgrade process; disables Bitlocker protection if enabled.
- Check - verify the upgrade was successful and that Bitlocker protection is re-enabled if it was disabled during the upgrade. DO NOT RUN MANUALLY - this is run via API when the upgrade process completes. It reports the upgrade status and re-enables Bitlocker protection if it was disabled by the upgrade process

Arguments

--Verify	Perform the pre-upgrade checks. Will perform the W11 Hardware Readiness check if the --Major argument is specified.

--Major	Performs a W10 to W11 Version upgrade instead of a same-version Build upgrade. Adds the Hardware Readiness Check to the validation process.
--Check	Perform the post-upgrade checks and re-enable Bitlocker protection if necessary. ***This is not intended to be run manually and is automatically scheduled by the install action to run 2 hours after starting the update process.***

Config File

None - all configuration is controlled centrally by MSP Builder.

Log File

RMUWPU.LOG

Special Features

Build Version Override

The **`HKLM\SOFTWARE\RMM : Latest_Bld`** key can be defined locally to override the MSP Builder Latest Build value. This allows early release testing during the 6-week period between release and MSPB updating the configuration settings. This will only work for values higher than the published MSPB setting. You cannot "update to" a specific version. ***Defining a lower than published value here will prevent upgrading beyond the locally-defined Build release!***

Controlling Automated Upgrading

The application can be run with --Verify to determine if the endpoint is ready for upgrading. If the verification is successful and --Major is NOT specified, the HKLM\SOFTWARE\RMM : AppRoleData will be written with "WUBC", representing "Windows Upgrade Build Compliance" status. Likewise, if --Major IS specified and all upgrade validations are successful, then "WUVC" will be written, representing "Windows Upgrade Version Compliance". The application can be run twice, once with and once without the --Major option to set both values as appropriate.

The MSP Builder Daily Audit will replicate this data to the SYSTEM ROLES custom field in the RMM. This allows (as an example) the following MSP Builder Daily Maintenance tasks to be used to control automated updating:

- Create a custom registry key, such as HKLM\SOFTWARE\MSP : AllowWxUpgrade and set the value to 0 (zero) to prevent upgrading.
- Create a custom Daily Maintenance task to run on Every Thursday to run the RMUWPU.BMS application with --Verify argument. Optionally define a second Verify task with the --Major option to allow Version upgrade testing.
- Create a custom Daily Maintenance task to run on the Last Friday of each month to display a message to the user that Windows will be upgraded. This task should use the appropriate WUBC or WUVC tag in the Roles option and a specific value in the AllowWxUpgrade registry key. You could use a value of 1 for Build upgrades and a value of 2 for Version upgrades if you desired.
- Create a custom Daily Maintenance task to run on the Last Friday of each month to run the RMUWPU.BMS application with no arguments, or with the --Major argument to allow W10 to W11 upgrades. This task should use the appropriate WUBC or WUVC tag in the Roles option and a specific value in the AllowWxUpgrade registry key. You could use a value of 1 for Build upgrades and a value of 2 for Version upgrades if you desired. This task should utilize the DelayUntil option to run at the end of the day.

By running the command weekly with the --Verify option, you can easily report on computers ready to upgrade. You can communicate with clients to prepare them for the upgrade. The day prior to the planned upgrade, run an RMM script to set the AllowWxUpgrade registry value to allow the upgrade through Daily Maintenance. When both the Registry Value and the System Roles allow the upgrade task, the task

will run. This allows the task to be scheduled monthly but only run when the AllowWxUpgrade registry is set. The registry value should be cleared to return full control over the upgrade task.

RMUW32 - Win-32 Time Service Tool

Corrects the configuration of the Windows Win-32 Time Service

This tool will update the Windows Time Service to be compliant with Microsoft's recommended best practices. It performs different actions on the PDCe compared to other member servers or domain controllers.

PDCe Actions

When run on a Domain Controller holding the PDC Emulator role, it will configure the time service to use NTP, and define three distinct public NTP hosts. By default, it uses 0-2.us.pool.ntp.org as time sources. The server will be configured to be *authoritative*, which will force all computers to remain in-sync with the PDCe even if it cannot sync with an external time source. If time should drift, all computers will drift together and avoid time related issues. *Microsoft recommends that ONLY the domain controller holding the PDCe role utilize NTP for time synchronization, and that it be configured as an authoritative source.* .

NOTE: If a GPS, Radio Clock, or other network time source is used, the configuration file or proper argument MUST be provided, or the service will be set to public NTP hosts.

Non-PDCe Actions

On all other computers, the W32Time service will be configured to utilize NT5DS as the domain time synchronization protocol. In this mode, all Domain Controllers will sync to the PDCe as the time master, and all other computers will sync with their local Domain Controller.

Platforms:

Runs on any RMM platform.

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--V	Verbose mode - report all actions
--T:l	List of alternate NTP hosts in semicolon-delimited format

Config File

RMUW32.INI

Log File

RMUW32_*#-DAY*.LOG

RMUWRT - Windows Reboot Tool

Performs intelligent reboots of workstations related to patching, or immediate reboots of any platform with a specified grace period

This application is usually invoked by the RMM Platform patching system, both before and after updating. The reboot will be performed under the control of the configuration file, depending on the arguments provided. It is also invoked by generic endpoint reboot scripts with the --G argument.

Only the --G argument is supported on servers. If --G is not specified, the tool will exit without any action on a server-class system (including a workstation defined with a Server role).

NOTE: The operation of this tool is fully documented in the Patching Operations Guide. Only the configuration options and core use will be documented here. **This is the only tool used by the RMM Suite to initiate a system reboot or shutdown.** All actions are logged, and the shutdown command writes appropriate reason codes and an "MSPB Initiated Reboot" message to the event log to accurately identify the source and reason for all system restarts.

Platforms:

Runs on any RMM platform.

Usage:

Invoked from the RMM platform or locally via the **RSRUN.BAT** script.

Arguments

--FORCE	Used with --G to force a reboot even if the -R reboot suppression is set in the patch code.
--G:#	Initiates a reboot on any platform with the specified grace period. All other arguments except --SHUT are ignored and the controls in PATCH.INI are not consulted unless --WP is also provided.
--SM	Reboot into Safe Mode with Networking for one cycle. Only valid with --G, and only when the RMM Suite is running in Autonomous mode. An "OTSMFlag" value is set in the registry. When the MAS starts, it disables the "Boot to Safe Mode" flag and clears that flag from the registry. Unrelated to the reboot tool itself, when the --SM option is invoked, a registry value "PRAction" can be loaded with a command that will be executed after rebooting and entering Safe Mode. This is usually a BAT file or other script to invoke multiple actions, ending with another reboot. This allows the automation of tasks while Safe Mode is active. This value must be defined before rebooting.
--SP | --WP	Sets Server|Workstation Patch Mode operation when used in conjunction with --G to reboot.
--SHUT	Valid only in combination with --G to perform a Shutdown instead of a reboot.
--SPR	Set the Patch Resume status flag (Azure Function). This controls whether patching will resume when the computer is powered back on if it was shut-down during the scheduled update cycle. This tool will display a message to inform the user when they log in that patching has begun.
--SET	Sets the Cycle Start timestamp to allow detection of post-update reboots and terminate nag messages.
--SR:<time>	Schedules a reboot for the specified time of the *current day*. A date cannot be specified, and the action will be ignored if the time defined has already passed.

--TERM	Terminates the patch nag process. Clears status and removes the scheduled task.
--PRE	Perform the pre-patch reboot if permitted / conditions allow. If reboots are enabled, the reboot is performed, but if reboots are not explicitly enabled, the reboot will be performed only if no user is logged in (console or RDP).
--POST[X]	Perform the post-patch reboot and cleanup actions. The POSTX option performs all cleanup actions but does not enforce the reboot.
--BL	Suppresses Bitlocker integration, requiring a PIN to be entered manually.
--N	Perform a Patch Nag process.

ONE argument from the above set is REQUIRED if not defined in the PR_Process registry key. Some RMM platforms will set the registry key to control actions rather than passing parameters. Using this tool with Flexible Patching does not require any configuration.

Config File

PATCH.INI - Controls the reboot actions related to pre and post-update rebooting of workstations. The following parameters are defined in the PATCHING section of this config file:

- PatchReboot — Allows reboot if this is true. All other settings are ignored if reboots are allowed!
- SchedReboot — Schedules the reboot to be performed at a specified time.
- NagMessage — The message text displayed when nagging for a post-patch reboot. This is only the FIRST message displayed - all other messages are defined in the language file for the RMM Suite User Interface, which can display locale-specific language messages.
- MaxNagStart — If defined, the time at which the nag message changes from a reminder to a countdown of hours before a forced reboot. 16:00 is the recommended value for this parameter.
- MaxNag — The maximum number of nag messages before forcing a reboot. This value MUST be less than the number of hours between MaxNagStart and midnight, which requires that the reboot occur no later than 11 PM. The recommended value is “6” when MaxNagStart=16:00.
- SuppressHourlyReminder — A “Y” or “N” parameter that, when enabled, will suppress the hourly nag messages and only display the countdown nag messages, starting at the MaxNagStart time. If MaxNagStart is undefined or invalid, this option is ignored. Default is disabled.
- NagAction — The default action to take when displaying a nag message, either Continue or Reboot - default is Continue.
- NagTimeout — How long the nag message will be displayed before taking the default action. Default is 30-minutes, which is also the maximum value.

This section applies only to workstations and controls the configuration of automated reboots vs. reboot nagging to delay the post-update reboot for some period of time.

Log File

RMUWRT_*#-DAY*.LOG

Administration

No administration is generally required for utility applications. Many are run automatically as part of regular automated operation, and others may be run in response to specific needs by a technician. Note that these tools can be initiated via RMM Platform scripts or directly from the computer's command-prompt by invoking the RSRUN.BAT script and specifying the application name and any necessary arguments.

Appendix I: List of RMM Suite Applications by Category

App ID	Name/Description	Category
RMMSEQ	RMM Maintenance Sequencing Engine	Maintenance
RMMSCU	System Cleanup Tool	Maintenance
RMMVDU	Volume Defrag Tool	Maintenance
RMMLSB	Local System Backup Tool	Maintenance
RMMCKD	Disk Health Check Tool	Maintenance
RMSSEQ	Smart Monitor Sequencing Engine	Smart Monitor
RMSSSC	System Security Status Check	Smart Monitor
RMSSAM	Server Health and Availability Check	Smart Monitor
RMSSBM	Server Boot Monitor	Smart Monitor
RMSICC	Internet Connection Failover Check	Smart Monitor
RMSDCC	Disk Capacity Check	Smart Monitor
RMSUSC	Local User Security Check	Smart Monitor
RMSNTP	Network Time Status and Configuration Check	Smart Monitor
RMASDU	System Data Utility / Audit Collection	Audit
RMUAAR	Audit User Admin Rights	Utilities
RMUOBA	Onboarding Automation	Utilities
RMUARN	Agent Rename Utility	Utilities
RMUAAC	Verify RMM Platform API Access	Utilities
RMUAMT	Third-Party App Management Utility	Utilities
RMUDFS	Disk Free Space tool	Utilities
RMUDVR	Disk Volume Reporting Tool	Utilities
RMUELC	Execute a Local Script (RS-Internal)	Utilities
RMUCSV	Get Cloud Script Variable	Utilities
RMUGES	Get and Execute a Script (RS-Internal)	Utilities
RMUGLZ	Get Diagnostic Data in Zip File	Utilities
RMUINI	INI-File Manipulation Tool	Utilities
RMUSCM	Manage MS Security Center	Utilities
RMUODJ	Offline Domain Join Utility	Utilities
RMUSPM	System Policy Manager	Utilities
RMUAIB	Agent Initialization and Branding	Utilities
RMURDT	Run Daily Tasks	Utilities
RMUSNAP	Collect System Process Snapshot	Utilities
RMUSUL	System User Logoff Tool	Utilities
RMUUAM	Manage Local User Accounts	Utilities
RMUWPU	Windows Platform Upgrade	Utilities
RMUW32	Win-32 Time Service Tool	Utilities
RMUWRT	Workstation Reboot Tool	Utilities