Progress®
Kemp® LoadMaster®

Technical Note PowerShell Breaking Changes

Generated on: 18 December 2024

Copyright

Visit the following page online to see Progress Software Corporation's current Product Documentation Copyright Notice/Trademark Legend: Product Documentation Copyright Notice & Trademarks | Progress

Table of Contents

Chapter 1: Introduction 4

Document Purpose 5

Intended Audience 5

Chapter 2: Object Structure. 6

Errors. 10

Chapter 3: Deprecated Cmdlets and Replacements 12

Chapter 4: References 23

1

Introduction

In version v7.2.39.0.334 of the Progress Kemp PowerShell module, a number of improvements were made. This included:

- Improving the output of the commands
- Renaming a number of commands to ensure the naming convention adheres to Microsoft guidelines
- Improvements to parameter values, such as changing some parameter value inputs from integer to string to better match the values in the LoadMaster Web User Interface (WUI)

As a result of these changes, the new version of the PowerShell module is not backwards compatible and scripts written for a version before v7.2.39.0.344 may not function correctly.

To check what version of the Progress Kemp PowerShell module you are using, run the following command:

```
Get-Module Kemp.LoadBalancer.Powershell
```

Related Links

- Document Purpose
- Intended Audience

Document Purpose

Document Purpose

This document outlines some of the major changes made to the PowerShell module. Being aware of these changes can help if you need to update any scripts written for older Progress Kemp PowerShell modules.

For further details on the Kemp PowerShell module, refer to the **PowerShell Interface Description** document on the Documentation page.

Intended Audience

Intended Audience

This document is intended to assist any users who may have written scripts for an older version of the Progress Kemp PowerShell module and now need to update these scripts to reflect a newer module version.

2

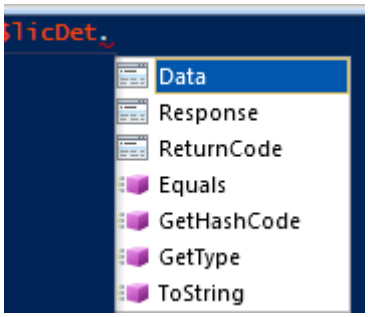
Object Structure

Object Structure

Before version 7.2.39.0.334 of the Progress Kemp PowerShell wrapper, the output format did not have a defined standard - each commandlet had variable structure and output.

In the 7.2.39.0.344 version of the wrapper, improvements were made to the output structure. All commands (except **Test-LmServerConnection**) return a PowerShell object with the following structure:

Note: The **Test-LmServerConnection** command returns **True** if the LoadMaster is reachable by the API, **False** if not (or if the API interface is disabled on the LoadMaster).



- ReturnCode (integer)
- Response (string)
- Data (PowerShell object, if any)

Note: As a result of the new object structure, the current Progress KempPowerShell wrapper is not compatible with scripts written based on an older version of the Progress Kemp PowerShell wrapper (before version 7.2.39.0.344).

The possible values for the ReturnCode field are:

- 200: The command completed successfully
- 4xx/500: The command ended with an error. The error code depends on the error type.

The possible values for the Response field are:

- Command successfully executed
- Description of the error when the command fails, for example **Unknown parameter value lmversion**.

The Data field contains the response, if any. The structure of this field depends on the command. The elements of this field can be accessed by assigning the command to a variable and using the “dot” notation (for example, **$lm_Version.Data.version** in the example below). The next argument is listed by PowerShell after you enter the dot, for example, **Data**, **Response**, or **ReturnCode**, depending on the data in the variable. The dot can be used multiple times to focus on certain information, for example, **$lm_Version.Data.version.Length**. If the command fails, this field is empty.

Example 1: Retrieve the installed LoadMaster firmware version:

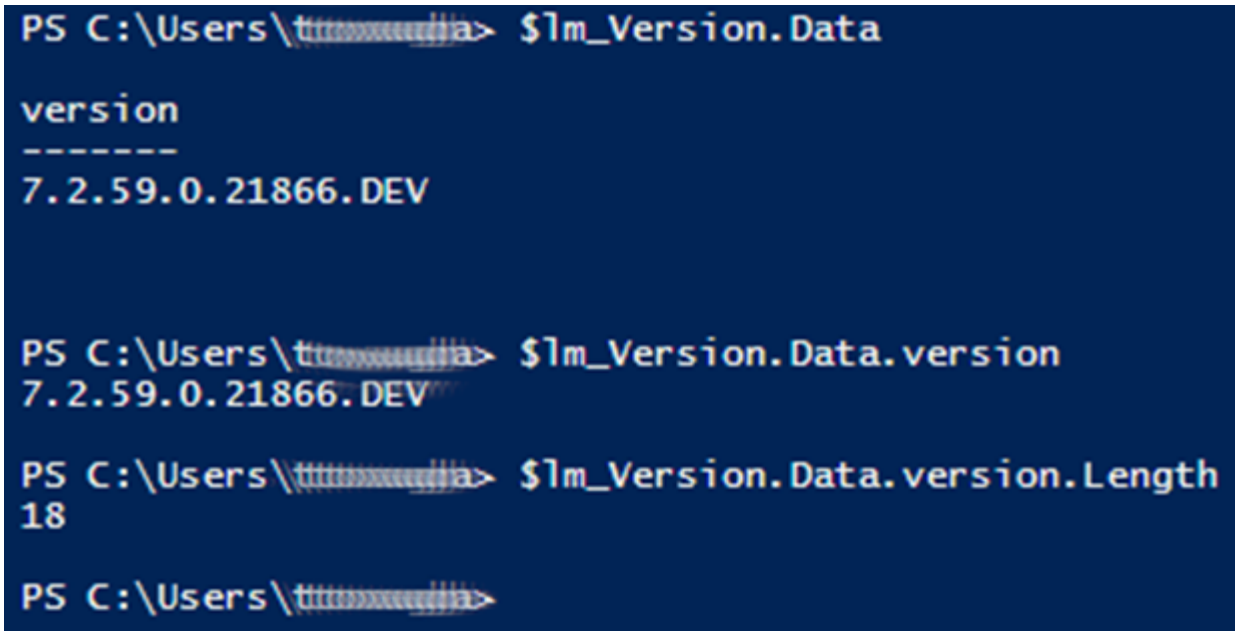
```
$lma = Get-LmParameter -Param version -LoadBalancer 172.21.59.189 -SubjectCN
user1

$lma | Format-List

ReturnCode : 200

Response   : Command successfully executed.

Data       : @{version=7.2.59.0.21866.DEV}
```




```
$lm_Version.Data.version

7.2.59.0.21866.DEV
```

Example 2: Retrieve the available licenses for a specific Order ID:

```
$licDet = Get-LicenseType -KempId jbloggs@kemptechnologies.com -Password
ExamplePassword -LoadBalancer 172.21.59.85 -Credential bal -OrderId
Example20170517

$licDet | Format-List

ReturnCode : 200

Response   : Command successfully executed.

Data       : @{License=System.Object[]}



$licDet.Data.License

id            : 0632b88b577c71591798268bcd4e01132f082309

name          : VLM-5000 ESP GEO with Basic 2 Years

available     : 1

description   : VLM-5000 ESP GEO with Basic 2 Years

tethered      : False
```

```
LicenseStatus : Permanent License

BuyMoreAt     : https://www.kemptechnologies.com/buy-me-now?KempID=
jbloggs@kemptechnologies.com

id            : fc488d991cffb7a5958625427d6bfb0b3edc008e

name          : VLM-5000 WAF GEO with Basic 3 Years

available     : 1

description   : VLM-5000 WAF GEO with Basic 3 Years

tethered      : False

LicenseStatus : Permanent License

BuyMoreAt     : https://www.kemptechnologies.com/buy-me-now?KempID=
jbloggs@kemptechnologies.com

name          : VLM-5000 with Enterprise Plus subscription

available     : 1

tethered      : 0

id            : 3eb92178611573946b422cf8d0df69d04c07fede

LicenseStatus : Temp License

description   : VLM-5000 with Enterprise Plus subscription

BuyMoreAt     : https://www.kemptechnologies.com/buy-me-now?KempID=
jbloggs@kemptechnologies.com
```

In the above example, the **$licDet.Data.License** is an array and each single element of the array can be accessed using the “[]” notation. For example, to access the field name of the second element of the previous array we have to use the following notation: **$licDet.Data.License[1].name**. The index of the array starts from 0. The NULL object is returned if we try to access a non-existing element.

The benefits of the structure of the command answers are:

- It is easy to check for success/error (ReturnCode)
- There is a short description (Response)
- The Data field returns a PowerShell object when successful and null when there is an error

Related Links

- Errors

Errors

Errors

If the error is a functional one (for example using the wrong credentials, parameter value or LoadMaster IP address) the cmdlet returns a PowerShell object, as described, with a ReturnCode containing the code of the error and with Response containing the description of the error that has occurred.

For example: Try to get the firmware version installed on the LoadMaster using an invalid certificate (installed in the Windows machine but not belonging to any user inside the LoadMaster):

```
$lma = Get-LmParameter -Param version -LoadBalancer 172.21.59.85 -SubjectCN
user1

$lma

ReturnCode Response                                                  Data

---------- --------                                                   ----

       401 The remote server returned an error: (401) Unauthorized.
```

If the error is due to a wrong/missing mandatory input, the cmdlet throws an exception. These types of errors do not return a ReturnCode because an exception has been thrown. The execution of the command is halted.

For example: Get the firmware version installed on the LoadMaster using a certificate not installed in the Windows machine:

```
$lma = Get-LmParameter -Param version -LoadBalancer 172.21.59.189 `

                                                        -SubjectCN
invalidcertificate

ERROR: Can't find a certificate with "invalidcertificate" as CN in the default
Cert:\CurrentUser\My store.

At C:\Users\ExampleUser\Work\Kemp.LoadBalancer.Powershell\
Kemp.LoadBalancer.Powershell.psm1:273 char:5

+                 Throw $errStr

+                 ~~~~~~~~~~~~~

    + CategoryInfo          : OperationStopped: (ERROR: Can't fi...tUser\My
store.:String) [], RuntimeException

    + FullyQualifiedErrorId : ERROR: Can't find a certificate with
"invalidcertificate" as CN in the default Cert:\CurrentUser\My store.
```

For example: Get the firmware version installed on the LoadMaster without credentials/certificate:

```
$lma = Get-LmParameter -Param version -LoadBalancer 172.21.59.189

ERROR: login method param is empty. Credentials or SubjectCN must be specified.

At C:\Users\ExampleUser\Work\Kemp.LoadBalancer.Powershell\
Kemp.LoadBalancer.Powershell.psm1:244 char:3

+         Throw $errStr

+         ~~~~~~~~~~~~~

    + CategoryInfo          : OperationStopped: (ERROR: login me...t be
specified.:String) [], RuntimeException

    + FullyQualifiedErrorId : ERROR: login method param is empty. Credentials
or SubjectCN must be specified.
```

Note: The connection drops if more than 30 calls are performed in less than 3 seconds over all API interfaces.

3

Deprecated Cmdlets and Replacements

Deprecated Cmdlets and Replacements

A number of commands have been deprecated and replaced with new commands with different names. Refer to the sections below for a list of deprecated cmdlets and replacements.

Note: These deprecated commands were removed completely in the 7.2.57 release.

Template Cmdlets

Deprecated cmdlet	Replacement cmdlet
ExportVSTemplate	Export-VSTemplate
UploadTemplate	Install-Template
DeleteTemplate	Remove-Template
ListTemplates	Get-Template

Logging Cmdlets

Deprecated cmdlet	Replacement cmdlet
Get-EmailOption	Get-LogEmailConfiguration
Set-EmailOption	Set-LogEmailConfiguration
Get-SyslogOption	Get-LogSyslogConfiguration
Set-SyslogOption	Set-LogSyslogConfiguration
Get-Statistics	Get-LogStatistics

SSO Cmdlets

Deprecated cmdlet	Replacement cmdlet
UploadRSAConfigurationFile	Install-SSORSAConfigurationFile
UploadRSANodeSecretAndPassword	Install-SSORSANodeSecretAndPassword
FlushSsoCache	Clear-SSOCache

Network Cmdlets

Deprecated cmdlet	Replacement cmdlet
ListIfconfig	Get-LmNetworkInterface
Get-NetworkOptions	Get-NetworkConfiguration
Set-NetworkOptions	Set-NetworkConfiguration
Get-DNSConfiguration	Get-NetworkDNSConfiguration
Set-DNSConfiguration	Set-NetworkDNSConfiguration
Update-LmDNSCache	Update-NetworkDNSCache
Get-SNMPOption	Get-NetworkSNMPConfiguration
Set-SNMPOption	Set-NetworkSNMPConfiguration
Get-Interface	Get-NetworkInterface
Set-Interface	Set-NetworkInterface
Add-InterfaceAddress	New-NetworkInterfaceAdditionalAddress
Remove-InterfaceAddress	Remove-NetworkInterfaceAdditionalAddress

Deprecated cmdlet	Replacement cmdlet
Get-Route	Get-NetworkRoute
New-Route	New-NetworkRoute
Remove-Route	Remove-NetworkRoute
Register-BondedInterface	Register-NetworkBondedInterface
Unregister-BondedInterface	Unregister-NetworkBondedInterface
Add-BondedInterface	New-NetworkBondedInterface
Remove-BondedInterface	Remove-NetworkBondedInterface
Add-Vlan	New-NetworkVLAN
Remove-Vlan	Remove-NetworkVLAN
Add-VxLan	New-NetworkVxLAN
Remove-VxLan	Remove-NetworkVxLAN

Application Delivery Cmdlets

Deprecated cmdlet	Replacement cmdlet
New-VirtualService	New-AdcVirtualService
Get-VirtualService	Get-AdcVirtualService
Set-VirtualService	Set-AdcVirtualService
Remove-VirtualService	Remove-AdcVirtualService
New-RealServer	New-AdcRealServer
Remove-RealServer	Remove-AdcRealServer
Set-RealServer	Set-AdcRealServer
Get-RealServer	Get-AdcRealServer
Enable-RealServer	Enable-AdcRealServer
Disable-RealServer	Disable-AdcRealServer
Remove-AdcVirtualServerRule	Remove-AdcVirtualServiceRule
Remove-VirtualServerRule	Remove-AdcVirtualServiceRule
New-RealServerRule	New-AdcRealServerRule
Remove-RealServerRule	Remove-AdcRealServerRule

Deprecated cmdlet	Replacement cmdlet
New-Rule	New-AdcContentRule
Remove-Rule	Remove-AdcContentRule
Set-Rule	Set-AdcContentRule
Get-Rule	Get-AdcContentRule
Get-L7Configuration	Get-AdcL7Configuration
Set-L7Configuration	Set-AdcL7Configuration
Get-LogSplitInterval	Get-AdcL7LogInsightSplitConfiguration
Set-LogSplitInterval	Set-AdcL7LogInsightSplitConfiguration
Get-ServiceHealth	Get-AdcServiceHealth
Set-ServiceHealth	Set-AdcServiceHealth
Add-NoCompressExtension	New-AdcHttpCompressionException
Remove-NoCompressExtension	Remove-AdcServiceHealth
Add-NoCacheExtension	New-AdcHttpCacheException
Remove-NoCacheExtension	Remove-AdcHttpCacheException
Get-AdaptiveCheck	Get-AdcAdaptiveHealthCheck
Set-AdaptiveCheck	Set-AdcAdaptiveHealthCheck
VSAddWafRule	New-AdcVsWafRule
VSRemoveWafRule	Remove-AdcVsWafRule
VSListWafRuleIds	Get-AdcVsWafRule

Security Cmdlets

Deprecated cmdlet	Replacement cmdlet
Set-AdminAccess	Set-SecAdminAccess
Get-WUIAuth	Get-SecWuiAuthentication
Set-WUIAuth	Set-SecWuiAuthentication
Get-WUISetting	Get-SecAdminWuiConfiguration
Set-WUISetting	Set-SecAdminWuiConfiguration
UserSetSystemPassword	Set-SecSystemUserPassword

Deprecated cmdlet	Replacement cmdlet
UserSetPermissions	Set-SecUserPermission
UserChangeLocalPassword	Set-SecUserPassword

System Cmdlets

Deprecated cmdlet	Replacement cmdlet
Initialize-LoadBalancer	Initialize-LmConnectionParameters
Test-ServerConnection	Test-LmServerConnection

Get/Set Cmdlets

Deprecated cmdlet	Replacement cmdlet
Get-AllParameters	Get-LmAllParameters
Get-Parameter	Get-LmParameter
Set-Parameter	Set-LmParameter

TLS Cmdlets

Deprecated cmdlet	Replacement cmdlet
New-Certificate	New-TlsCertificate
ListCert	Get-TlsCertificate
Remove-Certificate	Remove-TlsCertificate
Backup-Certificate	Backup-TlsCertificate
Restore-Certificate	Restore-TlsCertificate
New-IntermediateCertificate	New-TlsIntermediateCertificate
Remove-IntermediateCertificate	Remove-TlsIntermediateCertificate
GetCipherset	Get-TlsCipherSet
ModifyCipherset	Set-TlsCipherSet
DelCipherset	Remove-TlsCipherSet
HSMShow	Get-TlsHSM

Deprecated cmdlet	Replacement cmdlet
HSMConfigure	Set-TlsHSM
HSMGenerateClientCert	New-TlsHSMClientCert
HSMUploadCACert	Set-TlsHSMCACert

Web Application Firewall (WAF) Cmdlets

Deprecated cmdlet	Replacement cmdlet
ListWafRules	Get-WafRules
AddWafCustomData	New-WafCustomRuleData
DownloadWafCustomData	Export-WafCustomRuleData
DelWafCustomData	Uninstall-WafCustomRuleData
AddWafCustomRule	New-WafCustomRuleSet
DelWafCustomRule	Uninstall-WafCustomRuleSet
DownloadWafCustomRule	Export-WafCustomRuleSet
EnableWafRemoteLogging	Enable-WafRemoteLogging
DisableWafRemoteLogging	Disable-WafRemoteLogging
ListWafAuditFiles	Get-WafAuditFiles
DownloadWafAuditLog	Export-WafAuditLog
GetWafChangeLog	Export-WafChangeLog
ManInstallWafRules	Install-WafRulesDatabase
DownloadWafRules	Update-WafRulesDatabase
GetWafSettings	Get-WafRulesAutoUpdateConfiguration

GEO Cmdlets

Deprecated cmdlet	Replacement cmdlet
AddFQDN	New-GeoFQDN
Add-GeoFQDN	New-GeoFQDN
DeleteFQDN	Remove-GeoFQDN
ListFQDNs	Get-GeoFQDN

Deprecated cmdlet	Replacement cmdlet
ModifyFQDN	Set-GeoFQDN
AddCluster	New-GeoCluster
DeleteCluster	Remove-GeoCluster
ShowCluster	Get-GeoCluster
ModifyCluster	Set-GeoCluster
ClusterChangeLocation	Set-GeoClusterCoordinates
AddMap	New-GeoFQDNSiteAddress
DeleteMap	Remove-GeoFQDNSiteAddress
ModifyMap	Set-GeoFQDNSiteAddress
ChangeCheckerAddr	Set-GeoFQDNSiteCheckerAddress
AddCountry	Set-GeoFQDNSiteCountry
RemoveCountry	Remove-GeoFQDNSiteCountry
ChangeMapLocation	Set-GeoFQDNSiteCoordinates
AddCustomLocation	New-GeoCustomLocation
DeleteCustomLocation	Remove-GeoCustomLocation
ListCustomLocation	Get-GeoCustomLocation
EditCustomLocation	Set-GeoCustomLocation
AddIP	New-GeoIpRange
DeleteIP	Remove-GeoIpRange
ShowIP	Get-GeoIpRange
ListIPs	Get-GeoIpRange
ModifyIPLocation	Set-GeoIPRangeCoordinates
DeleteIPLocation	Remove-GeoIPRangeCoordinates
AddIPCountry	Set-GeoIPRangeCountry
RemoveIPCountryCustom	Remove-GeoIPRangeCountry
Remove-GeoIPRangeCustomLocation	Remove-GeoIPRangeCountry
RemoveIPCountry	Remove-GeoIPRangeCountry
AddIPCountryCustom	Set-GeoIPRangeCustomLocation

Deprecated cmdlet	Replacement cmdlet
ListMiscParameters	Get-GeoMiscParameter
ModifyMiscParameters	Set-GeoMiscParameter
LocationDataUpdate	Update-GeoDatabase
EnableGEO	Enable-LmGeoPack
DisableGEO	Disable-LmGeoPack
IsGEOEnabled	Test-LmGeoEnabled

Backup Cmdlets

Deprecated cmdlet	Replacement cmdlet
Backup-LoadBalancer	Backup-LmConfiguration
Restore-LoadBalancer	Restore-LmConfiguration
Get-BackupOption	Get-LmBackupConfiguration
Set-BackupOption	Set-LmBackupConfiguration

VPN Cmdlets

Deprecated cmdlet	Replacement cmdlet
CreateVpnConnection	New-LmVpnConnection
DeleteVpnConnection	Remove-LmVpnConnection
ListVpns	Get-LmVpnConnection
SetVpnAddrs	Set-LmVpnAddrs
SetVpnLocalIp	Set-LmVpnLocalIp
SetVpnLocalSubnets	Set-LmVpnLocalSubnet
SetVpnRemoteSubnets	Set-LmVpnRemoteSubnet
SetVpnSecret	Set-LmVpnSecret
StartVpnConnection	Start-LmVpnConnection
StopVpnConnection	Stop-LmVpnConnection
StartIkeDaemon	Start-LmVpnIkeDaemon
StopIkeDaemon	Stop-LmVpnIkeDaemon

Deprecated cmdlet	Replacement cmdlet
StatusIkeDaemon	Get-LmVpnIkeDaemonStatus
SetVpnPfsEnable	Set-LmVpnPfsEnable
SetVpnPfsDisable	Set-LmVpnPfsDisable

Add-on Cmdlets

Deprecated cmdlet	Replacement cmdlet
UploadAddon	Install-LmAddon
DeleteAddon	Remove-LmAddon
ListAddons	Get-LmAddOn

Patch Cmdlets

Deprecated cmdlet	Replacement cmdlet
Install-Patch	Install-LmPatch
Restore-Patch	Uninstall-LmPatch
Restart-LoadBalancer	Restart-Lm

Date/Time Cmdlets

Deprecated cmdlet	Replacement cmdlet
Get-DateTimeOption	Get-LmDateTimeConfiguration
Set-DateTimeOption	Set-LmDateTimeConfiguration

SDN Cmdlets

Deprecated cmdlet	Replacement cmdlet
AddSDNController	New-SdnController
DeleteSDNController	Remove-SdnController
ModifySDNController	Set-SdnController
GetSDNController	Get-SdnController

Application Front End (AFE) Cmdlets

Deprecated cmdlet	Replacement cmdlet
Get-AFEConfiguration	Get-LmAFEConfiguration
Set-AFEConfiguration	Set-LmAFEConfiguration
Update-IDSRule	Update-AFEIDSRules

Connection Limit Cmdlets

Deprecated cmdlet	Replacement cmdlet
AfeClientLimitList	Get-LmIPConnectionLimit
AfeClientLimitAdd	New-LmIPConnectionLimit
AfeClientLimitDelete	New-LmIPConnectionLimit

High Availability (HA) Cmdlets

Deprecated cmdlet	Replacement cmdlet
Get-HAOption	Get-LmHAConfiguration
Set-HAOption	Set-LmHAConfiguration
Set-AzureHAMode	Set-LmAzureHAMode
Get-AzureHAOption	Get-LmAzureHAConfiguration
Set-AzureHAOption	Set-LmAzureHAConfiguration
Set-AwsHAMode	Set-LmAwsHAMode
Get-AwsHAOption	Get-LmAwsHAConfiguration
Set-AwsHAOption	Set-LmAwsHAConfiguration

Diagnostic Cmdlets

Deprecated cmdlet	Replacement cmdlet
Get-DebugOption	Get-LmDebugConfiguration
Set-DebugOption	Set-LmDebugConfiguration
DoTcpDump	Trace-TcpTraffic

Cluster Cmdlets

Deprecated cmdlet	Replacement cmdlet
NMClusterStatus	Get-ClusterStatus
NMClusterCreate	New-Cluster
NMAddNode	New-ClusterNode
NMJoinCluster	Join-Cluster
NMEnableNode	Enable-ClusterNode
NMDisableNode	Disable-ClusterNode
NMDeleteNode	Remove-ClusterNode

4

References

References

Unless otherwise specified, the following documents can be found at https://docs.progress.com.

PowerShell, Interface Description