

Licenses

Last Modified on 04.16.26

Get-LiquitLicense

Synopsis

This command displays a list of all licenses known within Application Workspace.

Syntax

```
Get-LiquitLicense
[-LiquitContext <LiquitContext>]
[<CommonParameters>]

Get-LiquitLicense
[-ID] <guid[]>
[-LiquitContext <LiquitContext>]
[<CommonParameters>]

Get-LiquitLicense
[-EntityRef] <EntityRef[]>
[-LiquitContext <LiquitContext>]
[<CommonParameters>]

Get-LiquitLicense
[-Type] {Perpetual | SubscriptionMonthly | SubscriptionAnnual}
[-LiquitContext <LiquitContext>]
[<CommonParameters>]

Get-LiquitLicense
[-Search] <string>
[-LiquitContext <LiquitContext>]
[<CommonParameters>]

Get-LiquitLicense
[-Name] <string>
[-LiquitContext <LiquitContext>]
[<CommonParameters>]
```

Examples

This command displays all licenses known in Application Workspace:

```
Get-LiquitLicense
```

This command selects a specific license:

```
Get-LiquitLicense-id 00000000-0000-0000-0000-000000000000
```

Parameters

Recast

Name	Value	Description	Required	Default value
Search	<string>	This parameter serves a similar purpose to the search box found in the license table from the Application Workspace UI. It enables users to search multiple columns within the table, such as Name and Type. When a search term is provided, the parameter filters records by matching values in these indexed columns. However, it's important to note that not all columns in the table are indexed, meaning that searches for values in these non-indexed columns will not return results. Non-indexed columns include for example the ID and columns containing checkboxes.		
Name	<string>	Searches on the license name itself.		
LiquitContext	<LiquitContext>	Determines the selected zone.	No	Default

New-LiquitLicense

Synopsis

This command creates a new license.

Syntax

```
New-LiquitLicense
[-Name] <string>
[-Type {Perpetual | SubscriptionMonthly | SubscriptionAnnual}]
[-Count <int>]
[-Cost <string>]
[-Unit {Unspecified | Concurrent | User | Device}]
[-Description <string>]
[-LicenseKey <string>]
[-LiquitContext <LiquitContext>]
[-WhatIf]
[-Confirm]
[<CommonParameters>]
```

Example

This command creates a new license with the name "PowerShell License":

```
New-LiquitLicense -Name "PowerShell License" -Type Perpetual -Count 100 -Cost 100
```

Parameters

Name	Value	Description	Required	Default value
Name	<string>	Provide a name for the license.	Yes	
Type	{Perpetual SubscriptionMonthly SubscriptionAnnual}	The type of the license.	Yes	
Count	<int>	The number of licenses.	Yes	
Cost	<string>	The cost of the license.	Yes	
Enabled	<bool>	Determines whether or not the package is enabled.	No	True
Unit	{Unspecified Concurrent User Device}	Determines the unit of the license.	No	Unspecified
LicenseKey	<string>	The key associated with this license.		

Recast

Name	Value	Description	Required	Default value
Description	<string>	The description of the license.	No	
LiquitContext	<LiquitContext>	Determines the selected zone.	No	Default

Set-LiquitLicense

Synopsis

This command lets you edit the properties of a license.

Syntax

```
Set-LiquitLicense
[-License] <License[]>
[-Name <string>]
[-Type {Perpetual | SubscriptionMonthly | SubscriptionAnnual}]
[-Unit {Unspecified | Concurrent | User | Device}]
[-Count <int>]
[-Cost <string>]
[-Description <string>]
  [-LicenseKey <string>]
[-LiquitContext <LiquitContext>]
[-WhatIf]
[-Confirm]
[<CommonParameters>]
```

Example

The following script updates an existing license and renames it "Updated PowerShell License":

```
$license = Get-LiquitLicense -id 00000000-0000-0000-0000-000000000000
Set-LiquitLicense -License $license -Name "Updated PowerShell License"
```

Parameters

Name	Value	Description	Required	Default value
Name	<string>	Provide a name for the license.	No	
Type	{Perpetual SubscriptionMonthly SubscriptionAnnual}	The type of license.	No	
Count	<int>	The number of licenses.	No	
Cost	<string>	The cost of the license.	No	
Enabled	<bool>	Determines whether or not the package is enabled.	No	True
Unit	{Unspecified Concurrent User Device}	Determines the unit of the license.	No	Unspecified
LicenseKey	<string>	The key associated with this license.	No	
Description	<string>	The description of the license.	No	
LiquitContext	<LiquitContext>	Determines the selected zone.	No	Default

Remove-LiquitLicense

Recast

Synopsis

This command removes a license from the Application Workspace.

Syntax

```
Remove-LiquitLicense  
[-License] <License[]>  
[-LiquitContext <LiquitContext>]  
[-WhatIf]  
[-Confirm]  
[<CommonParameters>]
```

Example

The following script removes the selected license from the Application Workspace:

```
$license = Get-LiquitLicense -id 00000000-0000-0000-0000-000000000000  
Remove-LiquitLicense -License $license
```

Parameters

Name	Value	Description	Required	Default value
License	<License[]>	The license you wish to remove.	Yes	
LiquitContext	<LiquitContext>	Determines the selected zone.	No	Default

Get-LiquitLicensePackage

Synopsis

This command displays a list of all packages associated with a license.

Syntax

```
Get-LiquitLicensePackage  
[-License] <License[]>  
[-LiquitContext <LiquitContext>]  
[<CommonParameters>]
```

Example

The following script retrieves an Application Workspace license using a specific ID, then fetches the list of packages associated with that license:

```
$license = Get-LiquitLicense -id 00000000-0000-0000-0000-000000000000  
Get-LiquitLicensePackage -License $license
```

Parameters

Recast

Name	Value	Description	Required	Default value
License	<License>	The license for which you want to display the associated packages.	No	
LiquitContext	<LiquitContext>	Determines the selected zone.	No	Default

Add-LiquitLicensePackage

Synopsis

This command associates a package to a license.

Syntax

```
Add-LiquitLicensePackage  
[-License] <License>  
[-Package] <Package[]>  
[-LiquitContext <LiquitContext>]  
[-WhatIf]  
[-Confirm]  
[<CommonParameters>]
```

Example

The following script retrieves an Application Workspace license and a package by their IDs, then associates the package with the license

```
$license = Get-LiquitLicense -id 00000000-0000-0000-0000-000000000000  
$package = Get-LiquitPacakge -id 00000000-0000-0000-0000-000000000000  
Add-LiquitLicensePackage -License $license -Package $package
```

Parameters

Name	Value	Description	Required	Default value
License	<License>	The license to wich you want to add the package.	No	
Package	<Package[]>	The package that needs to be added to the license.	No	
LiquitContext	<LiquitContext>	Determines the selected zone.	No	Default

Remove-LiquitLicensePackage

Synopsis

This command disassociates all packages from a license.

Syntax

Recast

```
Remove-LiquitLicensePackage
[-License] <License>
[-Package] <Package[]>
[-LiquitContext <LiquitContext>]
[-WhatIf]
[-Confirm]
[<CommonParameters>]
```

Example

The following script removes the association between a license and a package:

```
$license = Get-LiquitLicense -id 00000000-0000-0000-0000-000000000000
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000
Remove-LiquitLicensePackage -License $license -Package $package
```

Parameters

Name	Value	Description	Required	Default value
License	<License>	The license associated with the package.	No	
Package	<Package[]>	The package that needs to be dissociated from the license.	No	
LiquitContext	<LiquitContext>	Determines the selected zone.	No	Default
