

## Filters

Last Modified on 04.16.26

### Get-LiquitFilter

### Synopsis

This command displays a list of filters belonging to a filter set.

### Syntax

```
Get-LiquitFilter
  [-FilterSet] <FilterSet>
  [-LiquitContext <LiquitContext>]
  [<CommonParameters>]
```

```
Get-LiquitFilter
  [-FilterSet] <FilterSet>
  [-Type] <string[]>
  [-LiquitContext <LiquitContext>]
  [<CommonParameters>]
```

```
Get-LiquitFilter
  [-FilterSet] <FilterSet>
  [-ID] <guid[]>
  [-LiquitContext <LiquitContext>]
  [<CommonParameters>]
```

```
Get-LiquitFilter
  [-FilterSet] <FilterSet>
  [-EntityRef] <EntityRef[]>
  [-LiquitContext <LiquitContext>]
  [<CommonParameters>]
```

### Examples

The following script lists all filters belonging to the specified filter set:

```
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000
$snapshot = Get-LiquitPackageSnapshot -id 00000000-0000-0000-0000-000000000000 -package $package
$filterset = Get-LiquitFilterSet -Snapshot $snapshot -id 00000000-0000-0000-0000-000000000000
$filter = Get-LiquitFilter -FilterSet $filterset
```

The following script will display all the filters belonging to the specified filter set of the Subnet type:

```
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000
$snapshot = Get-LiquitPackageSnapshot -id 00000000-0000-0000-0000-000000000000 -package $package
$filterset = Get-LiquitFilterSet -Snapshot $snapshot -id 00000000-0000-0000-0000-000000000000
$filter = Get-LiquitFilter -FilterSet $filterset -type subnet
```

### New-LiquitFilter

# Recast

## Synopsis

This command creates a new filter.

## Syntax

```
New-LiquitFilter
  [-FilterSet] <FilterSet[]>
  [-Type {Agent | AgentDeployment | AgentOffline | AgentType | AgentVersion | BrowserExists | ConditionSet | ContentScript | CurrentLanguage | DeviceFQDN | DeviceName | DirExists | DiskFreeSpace | EnvVarExists | EnvVarValue | FileExists | FileSize | FileVersion | Location | MacAppInstalled | MacAppInstalledVersion | MsiInstalled | MsiInstalledVersion | NetworkPing | PackageAvailable | PackageDistributed | PackageInstalled | PathExists | Platform | PlatformArch | PlatformLanguage | PlatformTimeZone | PlatformTimeZoneOffset | PlatformVersion | ProcessExists | ScriptStart | ServiceExists | ServiceState | SystemManufacturer | SystemModel | UrlSchemeExists | VarExists | VarValue | WinAppInstalled | WinAppInstalledVersion | WinAppStaged | WinAppStagedVersion | WmiQuery | WmiQueryCount}]
  [-Operator {Equal | NotEqual | Contains | NotContains | StartsWith | EndsWith | GreaterThan | LessThan}]
  [-Value <string>]
  [-Context {Device | User}]
  [-Settings <hashtable>]
  [-LiquitContext <LiquitContext>]
  [-WhatIf]
  [-Confirm]
  [<CommonParameters>]
```

## Example

The following script creates a new filter in the selected filter set:

```
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000
$snapshot = Get-LiquitPackageSnapshot -id 00000000-0000-0000-0000-000000000000 -package $package
$filterset = New-LiquitFilterSet -Snapshot $snapshot
new-LiquitFilter -FilterSet $filterset -type fileexists -Settings @{path = "C:\windows\settings.ini"} -Value "True"
```

## Parameters

Name	Value	Description	Required	Default value
FilterSet	<FilterSet[]>	The filter set where the filter needs to be added.	Entityfilter only	
Type	{Agent   AgentDeployment   AgentOffline   AgentType   AgentVersion   BrowserExists   ConditionSet   ContentScript   CurrentLanguage   DeviceFQDN   DeviceName   DirExists   DiskFreeSpace   EnvVarExists   EnvVarValue   FileExists   FileSize   FileVersion   Location   MacAppInstalled   MacAppInstalledVersion   MsiInstalled   MsiInstalledVersion   NetworkPing   PackageAvailable   PackageDistributed   PackageInstalled   PathExists   Platform   PlatformArch   PlatformLanguage   PlatformTimeZone   PlatformTimeZoneOffset   PlatformVersion   ProcessExists   ScriptStart   ServiceExists   ServiceState   SystemManufacturer   SystemModel   UrlSchemeExists   VarExists   VarValue   WinAppInstalled   WinAppInstalledVersion   WinAppStaged   WinAppStagedVersion   WmiQuery   WmiQueryCount}	The type of filter you want to apply.	Yes	

# Recast

Name	Value	Description	Required	Default value
Operator	{Equal   NotEqual   Contains   NotContains   StartsWith   EndsWith   GreaterThan   LessThan}	The operator used to evaluate the filter.	No	
Value	<string>	Determines whether the filter set should be true or not.	Yes	
Context	{Device   User}	Define when a package is available for a user/device.		Device
Settings	<hashtable>	The settings for the filter.		
LiquitContext	<LiquitContext>	Determines the selectedNo zone.		Default

## Set-LiquitFilter

### Synopsis

This command edits the properties of a filter.

### Syntax

```
Set-LiquitFilter
[-Filter] <Filter[]>
[-Operator {Equal | NotEqual | Contains | NotContains | StartsWith | EndsWith | GreaterThan | LessThan}]
[-Value <string>]
[-Context {Device | User}]
[-Settings <hashtable>]
[-LiquitContext <LiquitContext>]
[-WhatIf]
[-Confirm]
[<CommonParameters>]
```

## Remove-LiquitFilter

### Synopsis

This command removes a filter from the Application Workspace.

### Syntax

# Recast

```
Remove-LiquitFilter
[-Filter] <Filter[]>
[-LiquitContext <LiquitContext>]
[-WhatIf]
[-Confirm]
[<CommonParameters>]
```

## Example

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

```
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000
$snapshot = Get-LiquitPackageSnapshot -id 00000000-0000-0000-0000-000000000000 -package $package
$actionset = Get-LiquitActionSet -Snapshot $snapshot
$action = Get-LiquitAction -id 00000000-0000-0000-0000-000000000000 -ActionSet $actionset
$filterset = Get-LiquitFilterSet -Action $action -id 00000000-0000-0000-0000-000000000000
$filter = Get-LiquitFilter -FilterSet $filterset -id 00000000-0000-0000-0000-000000000000
Remove-LiquitFilter -Filter $filter
```

## Parameters

Name	Value	Description	Required	Default value
Filter	<Filter[]>	The filter(s) that should be removed.	Yes	
LiquitContext	<LiquitContext>	Determines the selected zone.	No	Default

## Further reading

---