

## Action Sets

Last Modified on 04.16.26

### Get-LiquitActionSet

### Synopsis

This command displays a list of all action sets belonging to a package snapshot, or selects a certain action set.

### Syntax

```
Get-LiquitActionSet
```

```
[-Snapshot] <PackageSnapshot>  
[-LiquitContext <LiquitContext>]  
[<CommonParameters>]
```

```
Get-LiquitActionSet
```

```
[-Snapshot] <PackageSnapshot>  
[-Type] {Distribute | Install | Launch | Uninstall | Repair}  
[-LiquitContext <LiquitContext>]  
[<CommonParameters>]
```

```
Get-LiquitActionSet
```

```
[-Snapshot] <PackageSnapshot>  
[-ID] <guid[]>  
[-LiquitContext <LiquitContext>]  
[<CommonParameters>]
```

```
Get-LiquitActionSet
```

```
[-Snapshot] <PackageSnapshot>  
[-EntityRef] <EntityRef[]>  
[-LiquitContext <LiquitContext>]  
[<CommonParameters>]
```

```
Get-LiquitActionSet
```

```
[-Search] <string>  
[-LiquitContext <LiquitContext>]  
[<CommonParameters>]
```

```
Get-LiquitActionSet
```

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

### Examples

The following script lists all available action sets of a package snapshot:

```
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000  
$snapshot = Get-LiquitPackageSnapshot -Package $package -id 00000000-0000-0000-0000-000000000000  
Get-LiquitActionSet -Snapshot $snapshot
```

The following script selects only the distribute type action sets of the package snapshot:

# Recast

```
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000  
$snapshot = Get-LiquitPackageSnapshot -Package $package -id 00000000-0000-0000-0000-000000000000  
Get-LiquitActionSet -Snapshot $snapshot -Type 'Distribute'
```

The following script selects the action set with the id 00000000-0000-0000-0000-000000000000:

```
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000  
$snapshot = Get-LiquitPackageSnapshot -Package $package -id 00000000-0000-0000-0000-000000000000  
Get-LiquitActionSet -Snapshot $snapshot -id 00000000-0000-0000-0000-000000000000
```

## Parameters

Name	Value	Description	Required	Default value
Snapshot	<PackageSnapshot>	The snapshot that should be retrieved. The type of the action set: <ul style="list-style-type: none"><li>Distribute – 0</li><li>Install – 1</li><li>Launch – 2</li><li>Uninstall – 3</li><li>Repair – 4</li></ul>		
Type	{Distribute   Install   Launch   Uninstall   Repair}		Yes	
Search	<string>	You can search by the name of the action sets, using a wildcard.		
Name	<string>	Searches on the action set name itself.		
LiquitContext	<LiquitContext>	Determines the selected zone.	No	Default

## New-LiquitActionSet

### Synopsis

This command creates a new action set.

### Syntax

```
New-LiquitActionSet  
[-Snapshot] <PackageSnapshot>  
[-Type] {Distribute | Install | Launch | Uninstall | Repair}  
[-Name <string>]  
[-Collection <string>]  
[-Enabled <bool>]  
[-Frequency {Always | OncePerUser | OncePerDevice}]  
[-Process {Sequential | SequentialIgnoreError | StopAtFirstEffectiveAction | StopAtFirstSuccessfulAction}]  
[-Position <int>]  
[-LiquitContext <LiquitContext>]  
[-WhatIf]  
[-Confirm]  
[<CommonParameters>]
```

### Example

The following script creates a new action set with the name "PowerShell ActionSet":

# Recast

```
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000
$snapshot = Get-LiquitPackageSnapshot -Package $package -id 00000000-0000-0000-0000-000000000000
New-LiquitActionSet -snapshot $snapshot -Type Install -name 'PowerShell ActionSet' -Enabled $true -Frequency Always -Process Sequential
```

## Parameters

Name	Value	Description	Required	Default value
Snapshot	<PackageSnapshot>	The snapshot that should get a new action set	Yes	
Type	{Distribute   Install   Launch   Uninstall   Repair}	The type of the action set: <ul style="list-style-type: none"><li>• Distribute</li><li>• Install</li><li>• Launch</li><li>• Uninstall</li><li>• Repair</li></ul>	Yes	
Name	<string>	Provide a name for the action set.	No	
Collection	<string>	The name of the shortcut for your action set. Available only for a Launch type action set.		
Frequency	{Always   OncePerUser   OncePerDevice}	Depending on the action set type there will be different options: <ul style="list-style-type: none"><li>• <i>Always</i> – the action will always be executed.</li><li>• <i>Once per user</i> – the action will be executed only once per user.</li><li>• <i>Once per device</i> – the action will be executed only once per device.</li></ul>		
Process	<string>	Determines the execution order of the actions within the action set: <ul style="list-style-type: none"><li>• <i>Sequential</i> – The actions in the action set are executed in order, stopping at the first error.</li><li>• <i>SequentialIgnoreError</i> – The actions in the action set are executed in order, ignoring errors.</li><li>• <i>StopAtFirstEffectiveAction</i> – The first effective action in the action set will be executed while the rest will be ignored.</li><li>• <i>StopAtFirstSuccessfulAction</i> – The first successful action in the action set will be executed while the rest will be ignored.</li></ul>	No	
Position	<int>	Application Workspace groups action sets by their type. You can change the position of an action set inside a group of action sets.	Yes	0
LiquitContext	<LiquitContext>	Determines the selected zone.	No	Default

## Move-LiquitActionSet

### Synopsis

This command moves the action set to a different position.

### Syntax

# Recast

```
Move-LiquitActionSet
[-ActionSet] <ActionSet[]>
[-Position] <int>
[-LiquitContext <LiquitContext>]
[-WhatIf]
[-Confirm]
[<CommonParameters>]
```

## Example

The following script moves an existing action set to position 1. For content or package reference actions, see the examples of [New-LiquitActionSet](#):

```
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000
$snapshot = Get-LiquitPackageSnapshot -package $package -id 00000000-0000-0000-0000-000000000000
$actionset = Get-LiquitActionSet -SnapShot $snapshot
Move-LiquitActionSet -ActionSet $actionset -position 1
```

## Parameters

Name	Value	Description	Required	Default value
ActionSet	<ActionSet[]>	The action set(s) that should be moved.	Yes	
Position	<int>	Determines the new position.	Yes	0
LiquitContext	<LiquitContext>	Determines the selected zone.	No	Default

## Set-LiquitActionSet

## Synopsis

This command edits the properties of a snapshot.

## Syntax

```
Set-LiquitActionSet
[-ActionSet] <ActionSet[]>
[-Name <string>]
[-Collection <string>]
[-Enabled <bool>]
[-Frequency {Always | OncePerUser | OncePerDevice}]
[-Process {Sequential | SequentialIgnoreError | StopAtFirstEffectiveAction | StopAtFirstSuccessfulAction}]
[-LiquitContext <LiquitContext>]
[-WhatIf]
[-Confirm]
[<CommonParameters>]
```

## Example

The following script updates an existing action set and renames it "Updated PowerShell Action Set":

# Recast

```
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000
$snapshot = Get-LiquitPackageSnapshot -package $package -id 00000000-0000-0000-0000-000000000000
$actionset = Get-LiquitActionSet -Snapshot $snapshot
Set-LiquitActionSet -ActionSet $actionset -name 'Updated PowerShell Action Set'
```

## Parameters

Name	Value	Description	Required	Default value
ActionSet	<ActionSet[]>	The action set or an array of action sets to be updated.	Yes	
Name	<string>	Provide a name for the action set(s).	No	
Enabled	<bool>	Determines whether or not the action set(s) will be enabled.	No	
Collection	<string>	The name of the shortcut for your action set. Available only for a Launch type action set. Depending on the action set type there will be different options:		
Frequency	{Always   OncePerUser   OncePerDevice}	<ul style="list-style-type: none"><li><i>Always</i> – the action will always be executed.</li><li><i>Once per user</i> – the action will be executed only once per user.</li><li><i>Once per device</i> – the action will be executed only once per device.</li></ul> Determines the execution order of the actions within the action set:		
Process	<string>	<ul style="list-style-type: none"><li><i>Sequential</i> – The actions in the action set are executed in order, stopping at the first error.</li><li><i>SequentialIgnoreError</i> – The actions in the action set are executed in order, ignoring errors.</li><li><i>StopAtFirstEffectiveAction</i> – The first effective action in the action set will be executed while the rest will be ignored.</li><li><i>StopAtFirstSuccessfulAction</i> – The first successful action in the action set will be executed while the rest will be ignored.</li></ul>	No	
LiquitContext	<LiquitContext>	Determines the selected zone.	No	Default

## Remove-LiquitActionSet

### Synopsis

This command removes an action set from the Application Workspace.

### Syntax

```
Remove-LiquitActionSet
[-ActionSet] <ActionSet[]>
[-LiquitContext <LiquitContext>]
[-Whatif]
[-Confirm]
[<CommonParameters>]
```

### Example

The following script retrieves an action set associated with a snapshot and deletes it:

# Recast

```
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000  
$snapshot = Get-LiquitPackageSnapshot -package $package -id 00000000-0000-0000-0000-000000000000  
$actionset = Get-LiquitActionSet -SnapShot $snapshot  
Remove-LiquitActionSet -ActionSet $actionset
```

## Parameters

Name	Value	Description	Required	Default value
ActionSet	<ActionSet[]>	The action set or an array of action sets to be removed.	Yes	
LiquitContext	<LiquitContext>	Determines the selected zone.	No	Default

---