Recast

Actions

Last Modified on 2026-11-02

Get-LiquitAction

This command displays a list of actions belonging to an action set.

Syntax

```
Get-LiquitAction
   [-ActionSet] <ActionSet>
   [-LiquitContext <LiquitContext>]
   [<CommonParameters>]

Get-LiquitAction
   [-ActionSet] <ActionSet>
   [-Type] <string[]>
   [-LiquitContext <LiquitContext>]
   [<CommonParameters>]

Get-LiquitAction
   [-ActionSet] <ActionSet>
   [-ID] <guid[]>
   [-LiquitContext <LiquitContext>]
   [<CommonParameters>]

Get-LiquitAction
   [-ActionSet] <ActionSet>
   [-EntityRef] <EntityRef[]>
   [-LiquitContext <LiquitContext>]
   [<CommonParameters>]

Get-LiquitAction
   [-ActionSet] <ActionSet>
   [-Search] <string>
   [-LiquitContext <LiquitContext>]
   [<CommonParameters>]

Get-LiquitAction
   [-ActionSet] <ActionSet>
   [-Name] <string>
   [-LiquitContext <LiquitContext>]
   [<CommonParameters>]
```

Examples

The following script lists all actions available within the selected action set:

```
$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
Get-LiquitAction -ActionSet $actionset
```

The following script selects only the “Install MSI” type action:

```
$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
Get-LiquitAction -ActionSet $actionset -Type 'pathmsiinstall'
```

The following script will display the selected action:

```
$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
Get-LiquitAction -ActionSet $actionset -id 00000000-0000-0000-0000-000000000000
```

Parameters

Name	Value	Description	Required	Default value
ActionSet	`<ActionSet[]>`	The actionset that should be retrieved.	No	
Type	`<string>`	The type of action that should be retrieved.	No	
Search	`<string>`	It enables users to search multiple columns within the table of the actions set from the Application Workspacew UI, 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.	No	
Name	`<string>`	Searches on the action name itself.	No	
LiquitContext	`<LiquitContext>`	Determines the selected zone.	No	Default

New-LiquitAction

This command creates a new action.

Syntax

```
New-LiquitAction
   [-ActionSet] <ActionSet[]>
   [-Name] <string>
   [-Type] <string>
   [-Enabled <bool>]
   [-IgnoreErrors <bool>]
   [-Settings <hashtable>]
   [-Context {Device | User | ElevatedUser | Server}]
   [-Position <int>]
   [-LiquitContext <LiquitContext>]
   [-WhatIf]
   [-Confirm]
   [<CommonParameters>]
```

Examples

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

```
$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
$action = New-LiquitAction -ActionSet $actionset -Name 'PowerShell Action' -Type 'pathmsiinstall' -Enabled $true -Ign
oreErrors $true -Settings @{path ='C:\';arguments = '/qn';transform =' ';}
```

The following script creates a new action for placing an uploaded file. Content actions require an attribute with a link to the content object:

```
$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
$action = New-LiquitAction -ActionSet $actionset -Name 'Place uploaded file' -Type 'contentcopy' -Enabled $true -Ign
oreErrors $true -Settings @{content = '4DF9E0F8.Netflix_6.95.602.0_x64__mcm4njqhnhss8.Appx'; destination ='c:\w
indows\temp';}
$content = New-LiquitContent -Path 'C:\install\4DF9E0F8.Netflix_6.95.602.0_x64__mcm4njqhnhss8.Appx' -FileName '
4DF9E0F8.Netflix_6.95.602.0_x64__mcm4njqhnhss8.Appx'
$attribute = New-LiquitAttribute -Entity $action -Link $content -ID 'content' -Settings: @{filename = '4DF9E0F8.Netfli
x_6.95.602.0_x64__mcm4njqhnhss8.Appx'}
```

The following script creates a new action for installing another package. Package reference actions require an attribute with a link to the package object:

```
$packageRef = Get-LiquitPackage -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
$actionset = Get-LiquitActionSet -Snapshot $snapshot
$action = New-LiquitAction -ActionSet $actionset -Name 'Install package' -Type 'installpackage' -Enabled $true -Ignor
eErrors $true -Settings @{title = '7-zip'; value = `$packageRef.ID`;}
$attribute = New-LiquitAttribute -Entity $action -Link $packageRef -ID 'package'
```



See Actions for a complete list of all possible action types.

Parameters

Name	Value	Description	Required	Default value
Name	`<string>`	Provide a name for the action.	Yes	
ActionSet	`<ActionSet[]>`	The actionset that should get a new action.	Yes	
Type	`<string>`	The type of action that should be added.	Yes	
Enabled	`<bool>`	Determines whether or not the action is enabled.	No	True
IgnoreErrors	`<bool>`	Determines whether or not the errors should be ignored.	No	False
Settings	`<hashtable>`	A hash table with the settings for the action.	No	
Position	`<int>`	Determines the new position	No	
Context	`{Device | User | ElevatedUser | Server}`	Defines the context in which the action should be executed: • Device • User • ElevatedUser • Server	No	Device
LiquitContext	`<LiquitContext>`	Determines the selected zone.	No	Default

Move-LiquitAction

This command moves the action to a different position.

Syntax

```
Move-LiquitAction
   [-Action] <Action[]>
   [-Position] <int>
   [-LiquitContext <LiquitContext>]
   [-WhatIf]
   [-Confirm]
   [<CommonParameters>]
```

Example

The following script moves the selected action to position 1. For content or package reference actions, see the examples of New-LiquitAction:

```
$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
$action = Get-LiquitAction -ActionSet $actionset
Move-LiquitAction -action $action -position 1
```

Parameters

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

---

Set-LiquitAction

This command edits the properties of an action.

Syntax

```
Set-LiquitAction
   [-Action] <Action[]>
   [-Name <string>]
   [-Enabled <bool>]
   [-IgnoreErrors <bool>]
   [-Settings <hashtable>]
   [-Context {Device | User | ElevatedUser | Server}]
   [-LiquitContext <LiquitContext>]
   [-WhatIf]
   [-Confirm]
   [<CommonParameters>]
```

Example

The following script updates the name of the selected action to "Updated PowerShell Action". For content or package reference actions, see the examples of New-LiquitAction:

```
$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
$action = Get-LiquitAction -ActionSet $actionset
Set-LiquitAction -action $action -name 'Updated PowerShell Action'
```

Parameters

Name	Value	Description	Required	Default value
Action	<Action[]>	The action(s) that should get a new action	Yes	
Enabled	<bool>	Determines whether or not the action is enabled	No	True
IgnoreErrors	<bool>	Determines whether or not the errors should be ignored	No	False
Name	<string>	Provide a name for the action	No	

Name	Value	Description	Required	Default value
Settings	`<hashtable>`	A Hashtable with the settings for the action	No	
Context	`{Device | User | ElevatedUser | Server}`	Defines the context in which the action should be executed: • Device • User • ElevatedUser • Server	No	Device
LiquitContext	`<LiquitContext>`	Determines the selected zone	No	Default

Remove-LiquitAction

This command removes an action from the Application Workspace.

Syntax

```
Remove-LiquitAction
   [-Action] <Action[]>
   [-LiquitContext <LiquitContext>]
   [-WhatIf]
   [-Confirm]
   [<CommonParameters>]
```

Example

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

```
$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
$action = Get-LiquitAction -ActionSet $actionset
Remove-LiquitAction -Action $action
```

Parameters

Name	Value	Description	Required	Default value
Action	`<Action[]>`	The action to be removed	Yes	
LiquitContext	`<LiquitContext>`	Determines the selected zone	No	Default
