Recast

Package Entitlements

Last Modified on 2026-11-02

Get-LiquitPackageEntitlement

This command displays a list of all entitlements associated with a package. Alternatively, you can use this command to select a specific entitlement.

Syntax

```
Get-LiquitPackageEntitlement
  [-Package] <Package>
  [-LiquitContext <LiquitContext>]
  [<CommonParameters>]

Get-LiquitPackageEntitlement
  [-Package] <Package>
  [-ID] <string[]>
  [-LiquitContext <LiquitContext>]
  [<CommonParameters>]
```

Examples

The following script displays all entitlements belonging to the specified package:

```
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000
Get-LiquitPackageEntitlement -Package $package
```

The following script selects a particular entitlement on the specified package:

```
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000
Get-LiquitPackageEntitlement -Package $package
```

New-LiquitPackageEntitlement

This command associates a new entitlement with a package.

Syntax

```
New-LiquitPackageEntitlement
   [-Package] <Package>
   [-Identity] <Identity>
   [-Stage {Development | Test | Acceptance | Production}]
   [-Publish {Hidden | Catalog | Workspace | Forced}]
   [-Icons <PackageEntitlementIcons>]
   [-Events <PackageEntitlementEvent[]>]
   [-LiquitContext <LiquitContext>]
   [-WhatIf]
   [-Confirm]
   [<CommonParameters>]

New-LiquitPackageEntitlement
   [-Package] <Package>
   [-Device] <Device>
   [-Stage {Development | Test | Acceptance | Production}]
   [-Publish {Hidden | Catalog | Workspace | Forced}]
   [-Icons  <PackageEntitlementIcons>]
   [-Events <PackageEntitlementEvent[]>]
   [-LiquitContext <LiquitContext>]
   [-WhatIf]
   [-Confirm]
   [<CommonParameters>]

New-LiquitPackageEntitlement
   [-Package] <Package>
   [-DeviceCollection] <DeviceCollection>
   [-Stage {Development | Test | Acceptance | Production}]
   [-Publish {Hidden | Catalog | Workspace | Forced}]
   [-Icons <PackageEntitlementIcons>]
   [-Events <PackageEntitlementEvent[]>]
   [-LiquitContext <LiquitContext>]
   [-WhatIf]
   [-Confirm]
   [<CommonParameters>]

New-LiquitPackageEntitlement
   [-Package] <Package>
   [-Context] <Context>
   [-Stage {Development | Test | Acceptance | Production}]
   [-Publish {Hidden | Catalog | Workspace | Forced}]
   [-Icons <PackageEntitlementIcons>]
   [-Events <PackageEntitlementEvent[]>]
   [-LiquitContext <LiquitContext>]
   [-WhatIf]
   [-Confirm]
   [<CommonParameters>]
```

Examples

The following script associates the specified identity with the specified package:

```
$identity = Get-LiquitIdentity -id "source\00000000-0000-0000-0000-000000000000"
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000
New-LiquitPackageEntitlement -Package $package -Identity $identity
```

The following script associates the specified identity with the specified package and publishes the package to the “Test” stage:

```
$identity = Get-LiquitIdentity -id "source\00000000-0000-0000-0000-000000000000"
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000
New-LiquitPackageEntitlement -Package $package -Identity $identity -Stage Test
```

The following script associates the specified identity with the specified package, with an auto-launch configured upon Application Workspace logon:

```
$identity = Get-LiquitIdentity -id "source\00000000-0000-0000-0000-000000000000"
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000
$event = New-Object Liquit.API.Server.V3.PackageEntitlementEvent
$event.Type = [Liquit.API.Server.V3.PackageEvent]::LiquitLogon
$event.ActionSet = [Liquit.API.Server.V3.ActionSetType]::Launch

$entitlement = New-LiquitPackageEntitlement -Package $package -Identity $identity -Events @($event)
```

Parameters

Name	Value	Description	Required	Default value
Package	`<Package>`	The package to which you want to associate a new entitlement.	Yes	
Identity	`<Identity>`	The identity you want to associate with the package.	Identity only	
Device	`<Device>`	The device to which you want to entitle this package.	Device only	
DeviceCollection	`<DeviceCollection>`	The device collection to which you want to entitle this package.	Device collection only	
Context	`<Context>`	The context to which you want to entitle this package.	Context only	
Stage	`{Development | Test | Acceptance | Production}`	The stage that should be published to the entitled identity.	No	Production
Publish	`{Hidden | Catalog | Workspace | Forced}`	How the package should be published to the entitled identity.	No	Catalog
Icons	`<PackageEntitlementIcons>`	The icons that should be published to the entitled identity (e.g. Desktop icons).	No	
Events	`<PackageEntitlementEvent[]>`	The events that should be active for the entitled identity (e.g. On refresh).	No	
LiquitContext	`<LiquitContext>`	Determines the selected zone.	No	Default

Set-LiquitPackageEntitlement

This command lets you edit the properties of an entitlement for a package.

Syntax

```
 Set-LiquitPackageEntitlement
[-Entitlement] <PackageEntitlement[]>
[-Stage {Development | Test | Acceptance | Production}]
[-Publish {Hidden | Catalog | Workspace | Forced}]
[-Approver <bool>]
[-Icons <PackageEntitlementIcons>]
[-Events <PackageEntitlementEvent[]>]
[-LiquitContext <LiquitContext>]
[-WhatIf]
[-Confirm]
[<CommonParameters>]
```

Example

The following script makes the specified identity an approver for the specified package:

```
$identity = Get-LiquitIdentity -id "source\00000000-0000-0000-0000-000000000000"
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000
$packageEntitlement  = Get-LiquitPackageEntitlement -Package $package -Identity $identity
Set-LiquitPackageEntitlement -Entitlement $packageEntitlement -Approver $true
```

Parameters

Name	Value	Description	Required	Default value
Entitlement	`<PackageEntitlement>`	The entitlement you want to edit.	Yes	
Stage	`{Development | Test | Acceptance | Production}`	The stage that should be published to the entitled identity.	No	Production
Publish	`{Hidden | Catalog | Workspace | Forced}`	How the package should be published to the entitled identity.	No	Catalog
Approver	`<bool>`	Gives the identity the rights to request and approve an application from **Catalog**.		false
Icons	`<PackageEntitlementIcons>`	The icons that should be published to the entitled identity (e.g. Desktop icons).	No	
Events	`<PackageEntitlementEvent[]>`	The events that should be active for the entitled identity (e.g. On refresh).	No	
LiquitContext	`<LiquitContext>`	Determines the selected zone.	No	Default

Remove-LiquitPackageEntitlement

This command removes an entitlement from a package.

Syntax

Recast

```
Remove-LiquitPackageEntitlement
  [-Entitlement] <PackageEntitlement[]>
  [-LiquitContext <LiquitContext>]
  [-WhatIf]
  [-Confirm]
  [<CommonParameters>]
```

Example

The following script removes the specified entitlement from the package:

```
$identity = Get-LiquitIdentity -id "source\00000000-0000-0000-0000-000000000000"
$package = Get-LiquitPackage -id 00000000-0000-0000-0000-000000000000
$packageEntitlement  = Get-LiquitPackageEntitlement -Package $package -Identity $identity
Remove-LiquitPackageEntitlement -Entitlement $packageEntitlement
```

Parameters

Name	Value	Description	Required	Default value
Entitlement	`<PackageEntitlement[]>`	The entitlement you wish to remove.	Yes	
LiquitContext	`<LiquitContext>`	Determines the selected zone.	No	Default