

# Restrict app installation only during OOB

Some application have the requirement to only be installed during the Autopilot provisioning.

Since Intune apps offers the possibility to run a PowerShell script before app installation, we are going to use this function to check whether the device is in OOB mode or not. This feature is referred to as “Requirement rule” and can be configured the follows:

## Instructions for replicating

Open or create an Intune app. Go to requirement rule and create a new rule with type “script”.

### Add a Requirement rule

Create a requirement.

|   |   |
|---|---|
| Requirement type *                              | <input type="text" value="Script"/>   |
| Script name *                                   | <input type="text" value="IsOOBComplete.ps1"/>  |
| Script file                                     | <input type="text" value="IsOOBComplete.ps1"/>  |
| Script content                                  | <pre>\$TypeDef = @"<br/><br/>using System;<br/>using System.Text;<br/>using System.Collections.Generic;<br/>using System.Runtime.InteropServices;</pre> |
| Run script as 32-bit process on 64-bit clients  | <input type="radio"/> Yes <input checked="" type="radio"/> No   |
| Run this script using the logged on credentials | <input type="radio"/> Yes <input checked="" type="radio"/> No   |
| Enforce script signature check                  | <input type="radio"/> Yes <input checked="" type="radio"/> No   |
| Select output data type *                       | <input type="text" value="Boolean"/>  |
| Operator *                                      | <input type="text" value="Equals"/>   |
| Value   | <input type="radio"/> Yes <input checked="" type="radio"/> No   |

Then enter all the necessary steps and configure the rule as in the screenshot.

# Script to report OOB status

This script return if the device currently is in OOB mode or not. Output "true" means the Autopilot mode is finished. Output "false" means, its currently in OOB / Autopilot mode.

```
$TypeDef = @"

using System;
using System.Text;
using System.Collections.Generic;
using System.Runtime.InteropServices;

namespace Api
{
    public class Kernel32
    {
        [DllImport("kernel32.dll", CharSet = CharSet.Auto, SetLastError = true)]
        public static extern int OOBEComplete(ref int blsOOBEComplete);
    }
}

"@

Add-Type -TypeDefinition $TypeDef -Language CSharp

$IsOOBEComplete = $false
$hr = [Api.Kernel32]::OOBEComplete([ref] $IsOOBEComplete)

$IsOOBEComplete
```

Original instructions and credits: [Detecting when you are in OOB – Out of Office Hours \(oofhours.com\)](https://oofhours.com)

---

Revision #1

Created 10 September 2024 19:06:09 by Luca Noah Caprez

Updated 10 September 2024 19:10:21 by Luca Noah Caprez