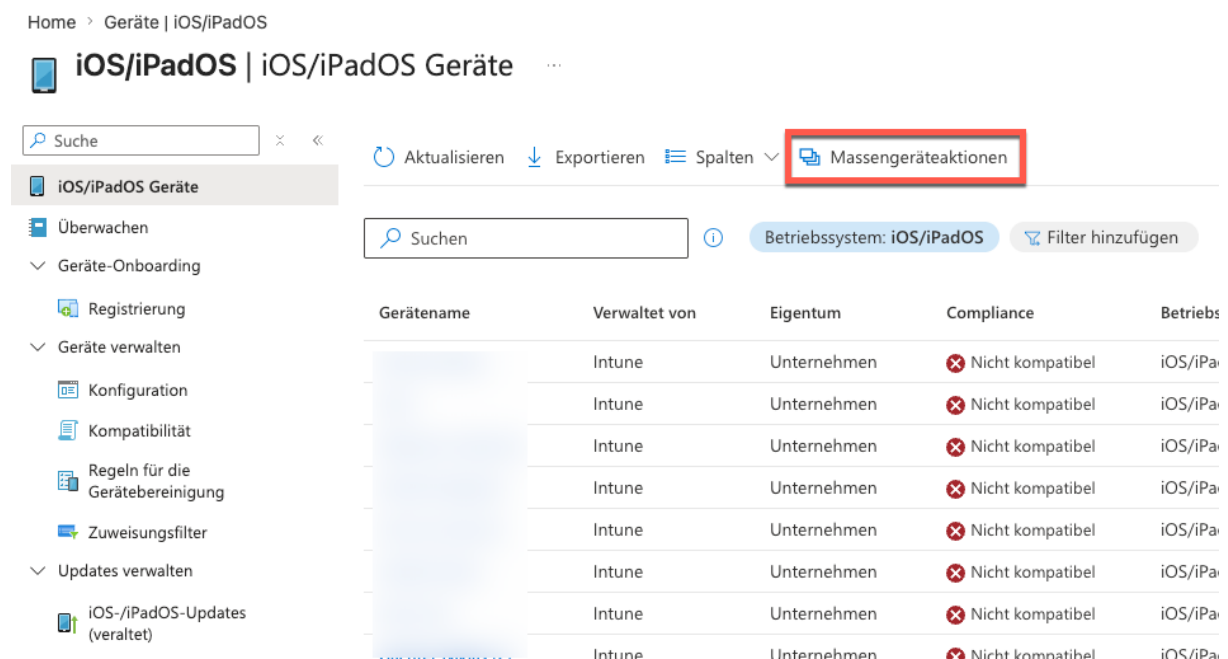


Vorarlberger Standardschulinstallation
Verfasser: Kuno Sandholzer

© 2025 IT-Regionalbetreuer Vorarlberg
6900 Bregenz, Römerstraße 14
Alle Rechte vorbehalten

© 2025 IT-Regionalbetreuer Vorarlberg
6900 Bregenz, Römerstraße 14
Alle Rechte vorbehalten



Danach Betriebssystem iOS/iPadOS und Geräteaktion Synchronisieren auswählen:

Home > Geräte | iOS/iPadOS > iOS/iPadOS | iOS/iPadOS Geräte

Massengeräteaktion ...

1 Grundlagen 2 Geräte 3 Überprüfen und erstellen

Betriebssystem *

iOS/iPadOS

Geräteaktion *

Synchronisieren



Intune wird versuchen, dieses Gerät einzuchecken. Falls der Vorgang erfolgreich ist, werden aktuelle Aktionen oder Richtlinien mit dem Gerät synchronisiert.

Danach am unteren Fensterrand Weiter auswählen und über Einzuschließende Geräte die Geräte wählen. Leider ist hier keine Gruppenauswahl möglich, sondern die Geräte müssen alle einzeln angeklickt werden.

Nach der Auswahl wieder auf Weiter und danach auf Erstellen klicken. Damit wird eine Synchronisation der ausgewählten Geräte angestoßen.

3 Massensynchronisierung via Powershell

Über ein Powershell-Skript kann die Synchronisierung genauer gesteuert werden. Das hier verwendete Programm synchronisiert wahlweise alle iPads oder nur die iPads von Mitgliedern einer bestimmten Sicherheitsgruppe.

Das Skript kann von https://download.vobs.at/wp-content/uploads/RBDownloads/Force_iPad_Sync/Force_iPad_Sync.ps1 heruntergeladen werden.

3.1 Synchronisierung aller iPads anstoßen

Wird das Skript ohne Parameter aufgerufen, so stößt es eine Synchronisierung aller iPads an:

```
.\Force_iPad_Sync.ps1
```

3.2. Synchronisierung der iPads einer Usergruppe

Dem Skript kann alternativ eine Sicherheitsgruppe als Parameter übergeben werden. Das Skript ermittelt dann die Mitglieder dieser Gruppe, anschließend welche iPads diesen Benutzern zugeordnet sind und schickt dann an diese Geräte eine Synchronisierungsaufforderung.

```
.\Force_iPad_Sync.ps1 -GroupName grp1a
```

3.3 Ablauf des Skripts

Nach dem Start prüft das Skript, ob die aktuelle Version von Microsoft.Graph installiert ist. Falls die Module fehlen oder ein Update benötigen, so werden sie zuerst installiert bzw. aktualisiert. Dies kann eine Weile dauern.

Danach fragt das Skript nach den Anmeldedaten für den Tenant. Zur Benutzung ist die Anmeldung als Tenant-Administrator oder MDM-Administrator notwendig.

3.4 Skriptcode

```
# Force_iPad_Sync.ps1
# Author: Kuno Sandholzer, sandholzer@vobs.at
# Version: 1.1
# April 2026
#

param(
    [string]$GroupName
)

$moduleName = "Microsoft.Graph"

$installed = Get-Module -ListAvailable -Name $moduleName |
    Sort-Object Version -Descending |
    Select-Object -First 1

$latest = Find-Module -Name $moduleName

if (-not $installed) {
    # Modul fehlt komplett
    Install-Module $moduleName -Scope CurrentUser -Force
}
elseif ($installed.Version -lt $latest.Version) {
```

```

        # Update notwendig
        Update-Module $moduleName -Force
    }
    else {
        Write-Host "Modul ist aktuell $($installed.Version)"
    }

    Connect-MgGraph -Scopes @(
        'DeviceManagementManagedDevices.ReadWrite.All',
        'DeviceManagementManagedDevices.Read.All',
        'Group.Read.All'
    )

    try {
        $allDevices = Get-MgDeviceManagementManagedDevice -All -ErrorAction Stop
    }
    catch {
        $_ | Format-List -Force
    }

    if (-not $GroupName) {

        Write-Host 'Keine Gruppe angegeben - synchronisiere alle iOS-Geräte'

        foreach ($device in $allDevices) {
            if ($device.OperatingSystem -eq 'iOS') {
                Sync-MgDeviceManagementManagedDevice -ManagedDeviceId $device.Id
                Write-Host ('Sync request sent to: ' + $device.DeviceName)
            }
        }
    }
    else {

        Write-Host ('Gruppe ' + $GroupName + ' wird verwendet...')

        $filter = "displayName eq '$GroupName'"
        $group = Get-MgGroup -Filter $filter

        if (-not $group) {
            Write-Error ('Gruppe ' + $GroupName + ' nicht gefunden!')
            Disconnect-MgGraph
            return
        }

        $groupMembers = Get-MgGroupMember -GroupId $group.Id -All

        $userIds = @()
        foreach ($member in $groupMembers) {

```

```
        $userIds += $member.Id
    }

    # DEBUG Line: Write-Host('User IDs: ' + $userIds);

    $devicesToSync = @()

    foreach ($device in $allDevices) {
        if ($device.OperatingSystem -eq 'iOS' -and $userIds -contains
$device.UserId) {
            $devicesToSync += $device
        }
    }

    foreach ($device in $devicesToSync) {
        Sync-MgDeviceManagementManagedDevice -ManagedDeviceId $device.Id
        Write-Host ('Sync request sent to: ' + $device.DeviceName)
    }
}

Disconnect-MgGraph
```

★Erstellt: 2026-04-30 05:27:53 UTC von Kuno Sandholzer

✎Zuletzt aktualisiert: 2026-04-30 12:10:57 UTC von Kuno Sandholzer