Lenovo Device Manager (LDM)
User Guide

Name of Document	LDM User Guide
Version Number	V2.4
Release Date	1 November 2022
Modifications/Additions	1. Bulk Power Management for Intel vPro® devices 2. Remote Desktop Management for Intel vPro devices: Full Screen 3. Remote Desktop Management for Intel vPro devices: CTL-ALT-DEL 4. Support for AR/VR devices 5. Device Type categorization on Device Tray 6. Update files included in Windows device provisioning package

Contents

1 OVERVIEW

As the demand for more devices grows and the move to the cloud continues, Lenovo Device Manager provides a flexible, scalable endpoint and app management solution for any Lenovo Windows or Android device.

LDM features include:

- Robust device details and health status
- Simplified device & cloud-based application updating
- Integration with Intel vPro® EMA
- Quick deployment of software and add-on services
- Better end-user experience
- Safe, secure platform

2 Setup & Configuration

2.1 Organization Setup

2.2 Manage Organization

2.3 User Preferences

2.4 User Management

2.5 User Groups

3 Manage Devices

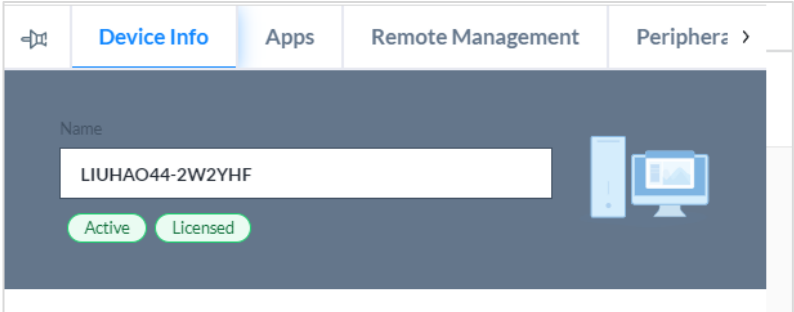
Devices represent the various device types that are in your organization and typically used by employees. A device type can fall under any of the following categories:

	Current Device Type Categories			
	PCs	**SmartEdge**	**Tablet / Mobile**	**AR / VR**
Examples	Any Lenovo notebook, desktop, workstation, etc	Any Lenovo edge appliance and servers	Any Lenovo tablets or mobile device	Any Lenovo AR or VR appliance

3.1 Add Devices

3.2 Manage Devices

3.3 View Device Information and Perform Basic Actions

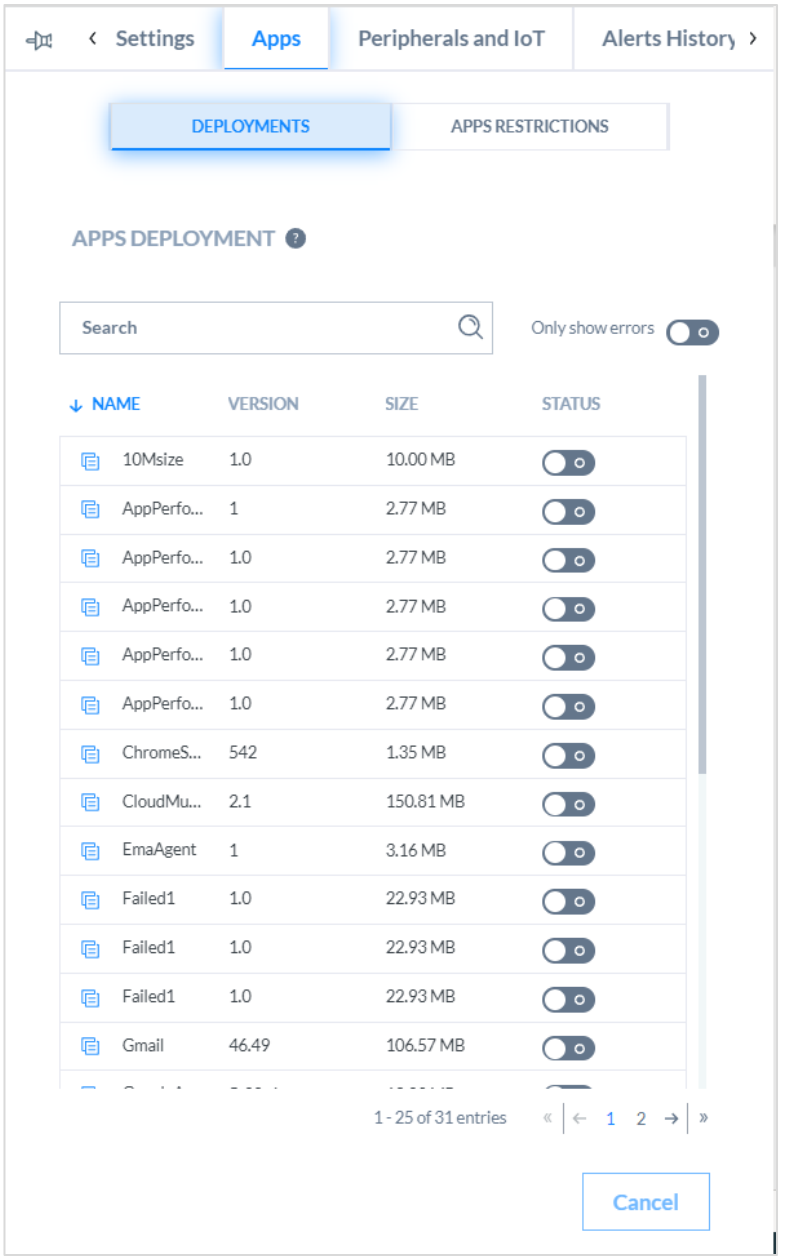


- No adjust volume
- No airplane mode
- No ambient display
- No control apps
- No autofill
- No Bluetooth
- No Bluetooth sharing
- No config Bluetooth
- No config brightness
- No config cell broadcasts
- No config credentials
- No config date time
- No config locale
- No config location
- No config mobile networks
- No config private DNS
- No config screen timeout
- No config tethering
- No config VPN
- No config WiFi
- No content capture
- No content suggestions
- No create windows
- No cross-profile copy-paste
- No factory reset
- No fun
- No install apps
- No install unknown sources
- No install unknown sources globally
- No modify accounts
- No physical media
- No network reset
- No outgoing beam
- No outgoing calls
- No printing
- No remove user
- No safe boot
- No set user icon
- No set wallpaper
- No sharing into profile
- No share location
- No SMS
- No system error dialogs
- No unified password
- No uninstall apps
- No unmute microphone
- No file transfer through USB
- No user switch
- No apps verification

***Note**: In LDM, hovering over tool tip next to each Restriction provides its definition*

The following options are available for a user on the device tray – ***Apps*** tab:

- ***Deployments*** sub-tab:
    - View LDM-managed applications (software, firmware, driver) on device
    - Deploy application updates to the device
    - Uninstall applications from the device
    - View deployment status

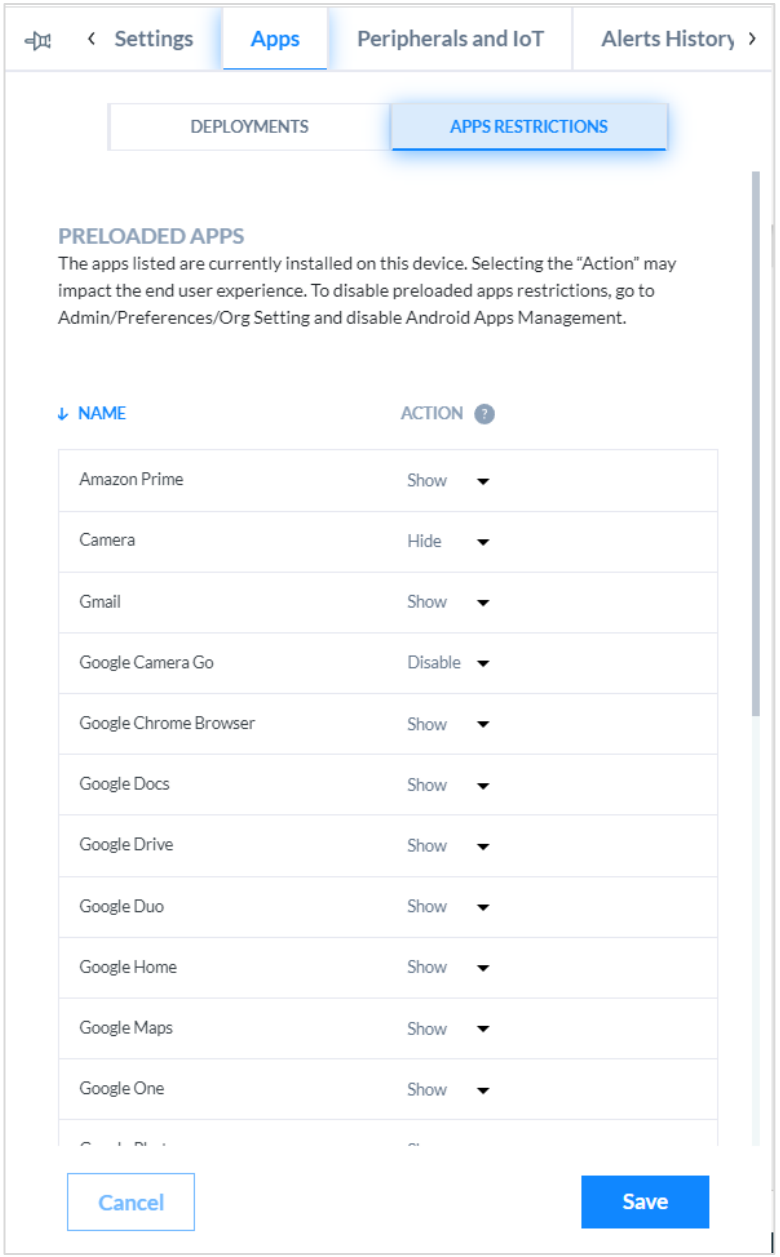


- ***App Restrictions*** sub-tab:

  Note: This feature is currently only available for Android devices (tablets) and **MUST BE ENABLED** in Org Settings (see Section 2.2.2 for more information)

  - Displays list of standard pre-loaded Android apps
  - Allows end user app experience to be controlled remotely

  1. Find the App to be managed
  2. Under "Action", click the drop down to select from the following Actions:
     - **Show** (default) – allows app to be visible and usable on the device
     - **Hide** – app will be hidden on the device and unusable
     - **Disable** – app will be seen, but will be disabled and unusable

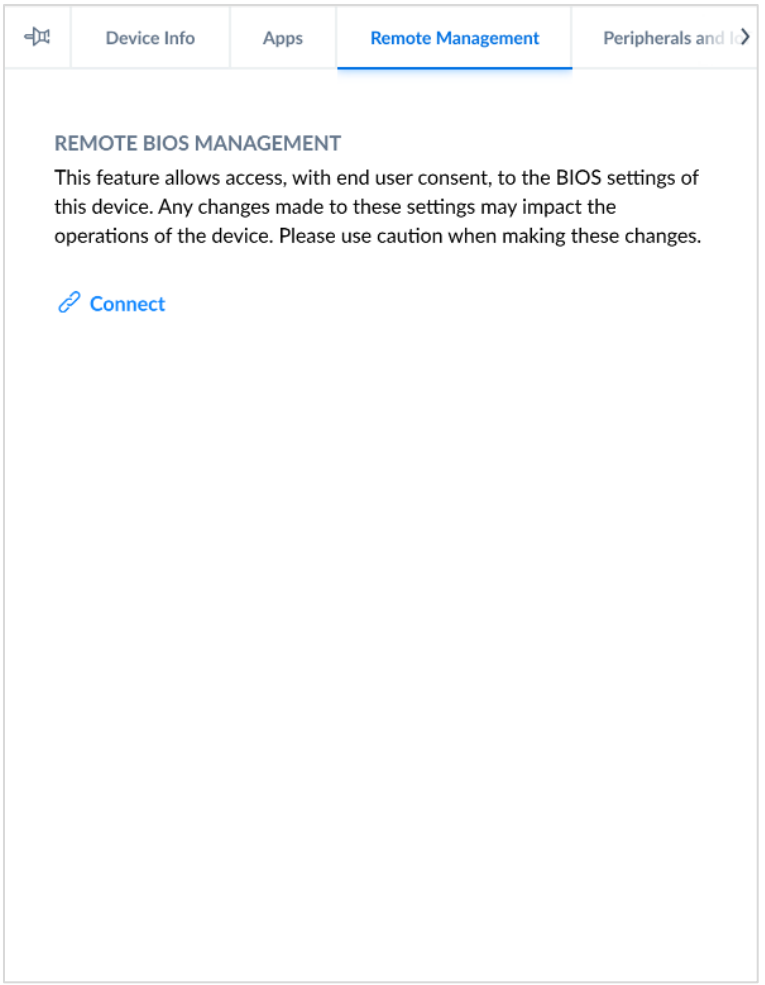


The following options are available for a user on the device tray – ***Remote Management*** tab:

Note: This feature is currently only available for Windows devices (with Intel vPro agent installed)

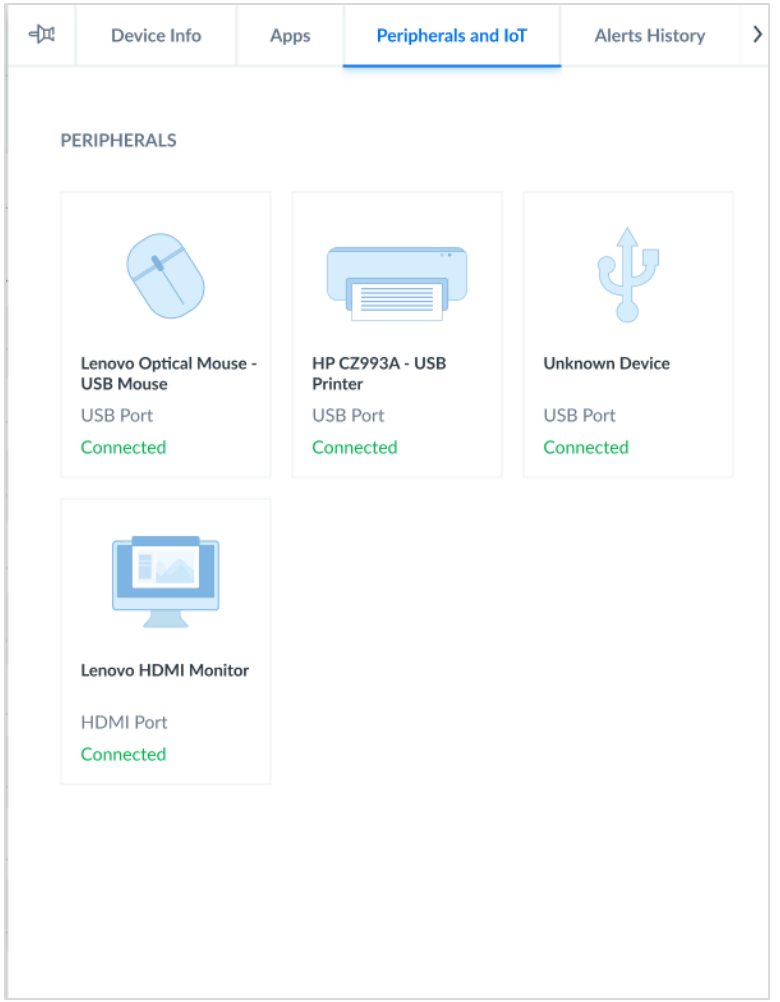
- Remote BIOS Management: Allows IT Admins to access the device's BIOS settings
- Requires Intel vPro agent installed on the device

For more information on Remote BIOS Management, see Intel vPro Support, section 3.4



The following options are available for a user on the device tray – ***Peripherals and IoT*** tab:

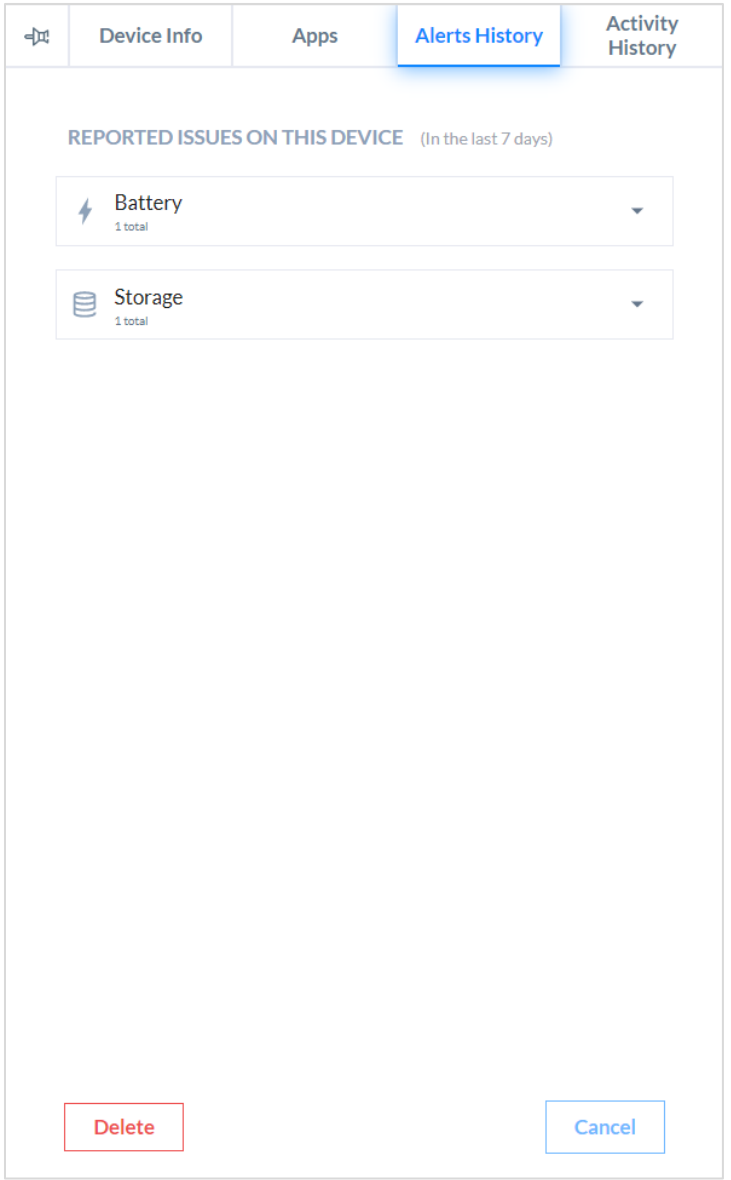
- View any USB or HDMI connected peripheral connected to the device
- See port type in use



The following options are available for a user on the device tray – ***Alert History*** tab:

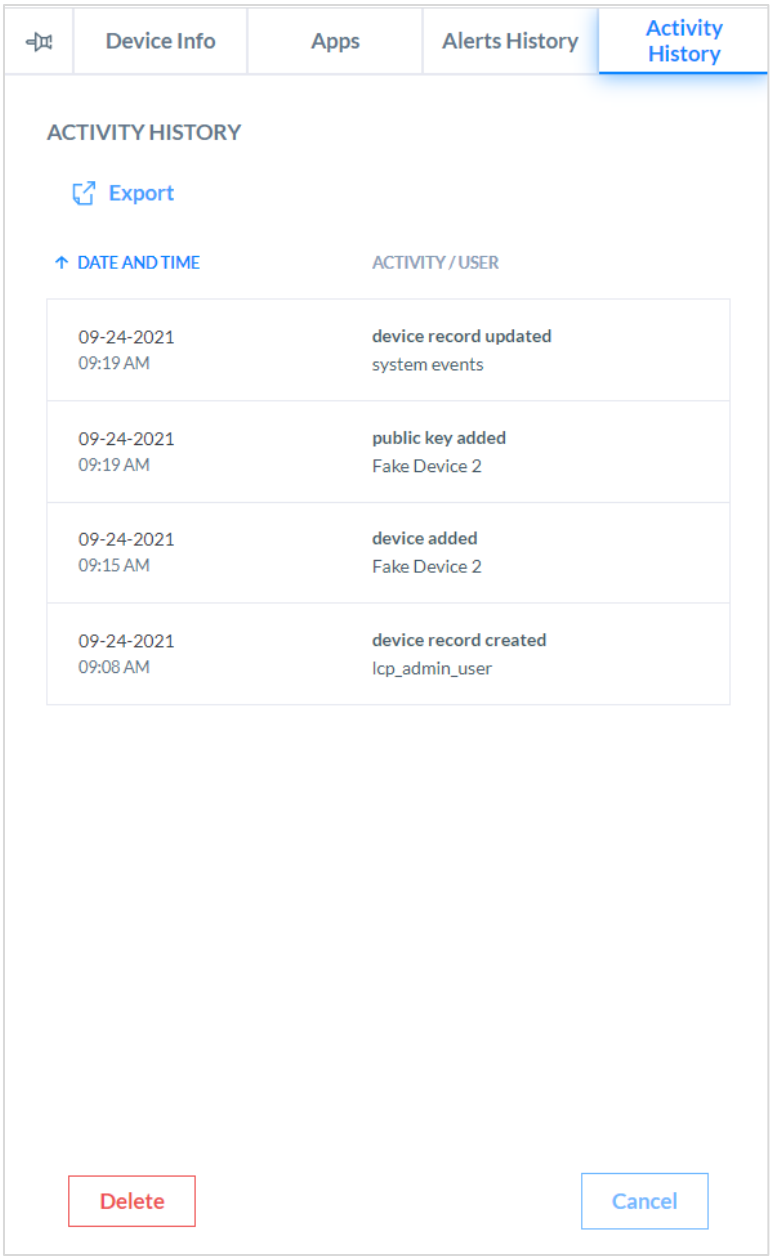
- View any “Low Battery” alert
- View any “Storage” alert
- View any “OTA Deployment” alert
- Delete device

NOTE: Alert status is reflective over a rolling seven-day period.



The following options are available on the device tray - ***Activity History*** tab:

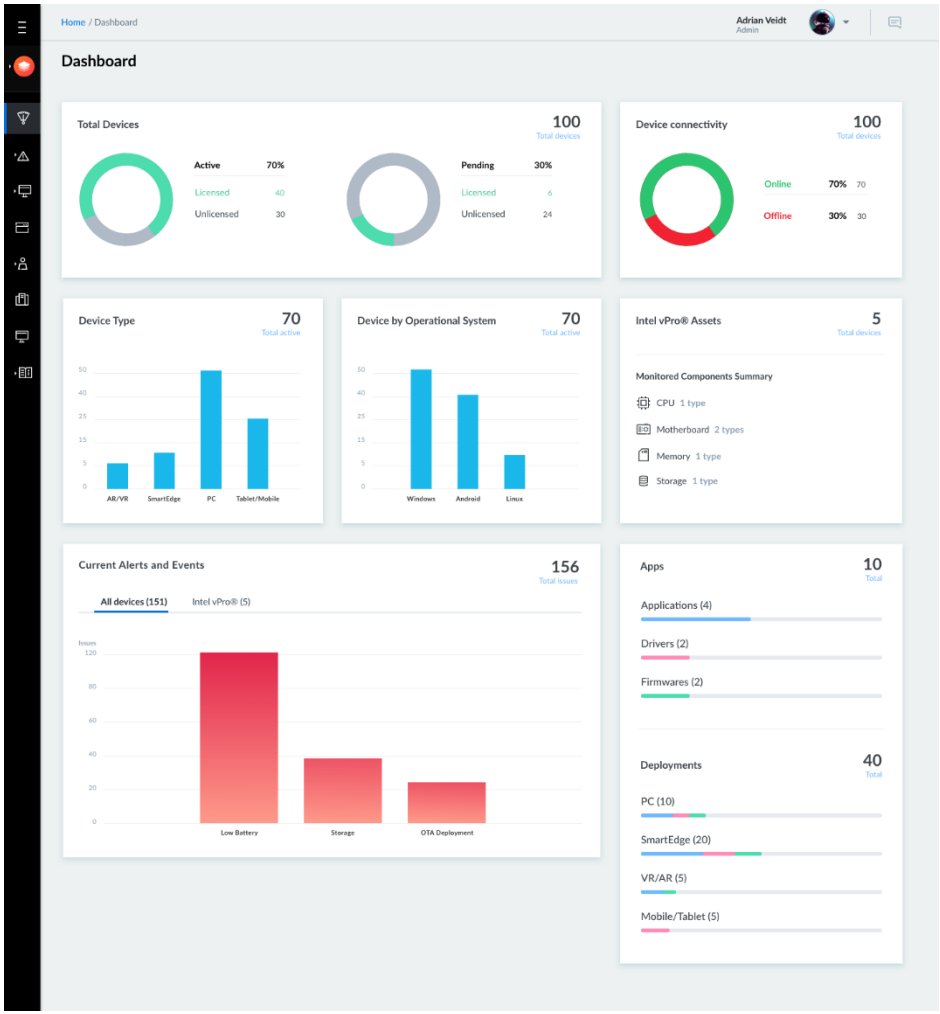
- View the device Activity History
- Export device Activity History to CSV file
- Delete device



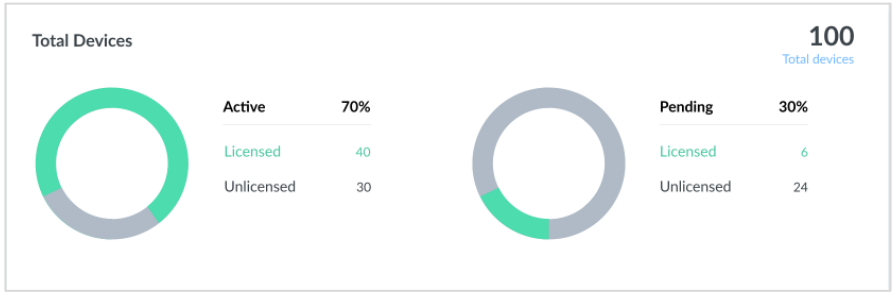
6 Lenovo Device Manager Dashboard

The Dashboard is the home page for Lenovo Device Manager and offers an at-a-glance overview of the devices in your organization. The Dashboard consists of several widgets, where each widget represents different device management categories.

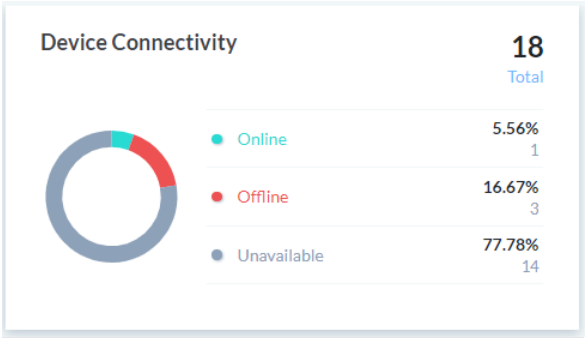
Clicking on metrics displayed on a chart will typically navigate the user to the corresponding detail pages throughout the portal. This data is updated throughout the day.



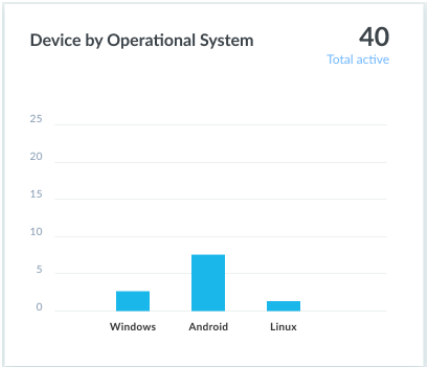
Dashboard Widgets

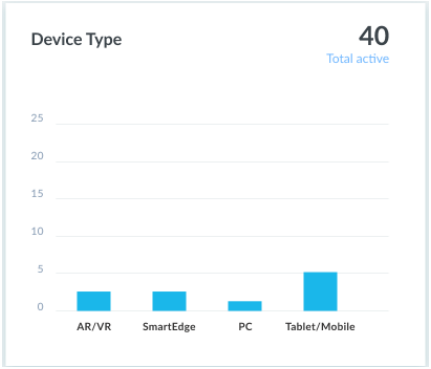


The total devices claimed on Lenovo Client Remote Management, highlighting licensing status. Clicking on Active or Pending charts will automatically take you to the Device list, filtered by the status selected.

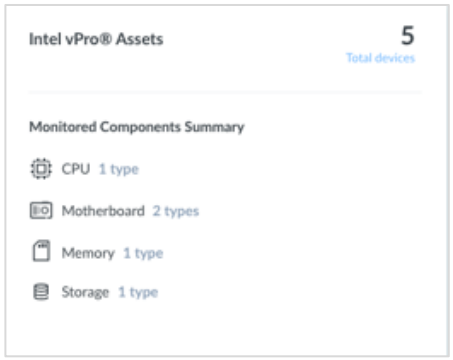


Current connectivity status, highlighting devices that are currently online or offline. Unavailable devices have not yet been fully claimed on LDM.

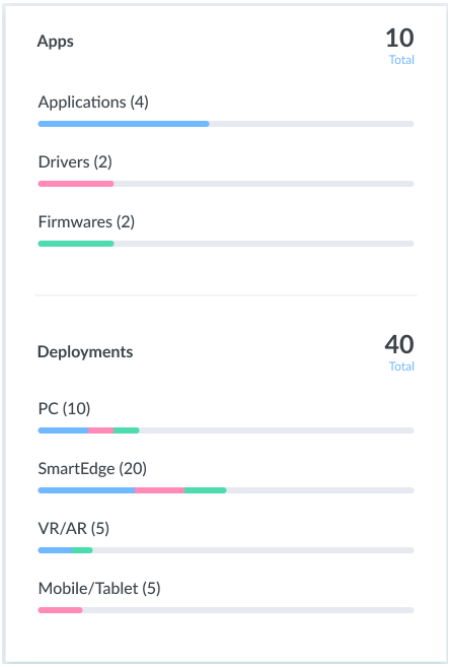




Breakdown of devices by “Device Type” and “Operating System” allows you to track the number of each being managed through LDM.



Summarizes the Intel vPro® device assets, including all monitored hardware components.



Provides an overview of the different app types being managed through LDM as well as how those apps are being to deployed.

Provides information on any alerts detected over the last 7 days as of the last data update. Clicking on a category will navigate you to the issue report for that respective category, listing impacted devices. Clicking on the Intel vPro® tab will show asset change alerts. See the “Reports” section for more details on each.

7 REPORTS

7.1 Low Battery Report

7.2 Low Storage Report

7.3 OTA Deployment Report

7.4 Decommissioned Devices Report

7.5 Intel vPro® Asset Changes

8 TROUBLESHOOTING - FAQ

Question:	**I am unable to login to the portal; my username or password is incorrect.**
Answer:	Your login credentials must match the login setup in Lenovo ID/Lenovo Passport. If you are still having problems logging in, reset the Lenovo ID password and try again.
Question:	**During device claiming (provisioning), I am asked to run a 'PowerShell' script. However, I am getting a " UnauthorizedAccess" message. What should I do?**
Answer:	To execute the PowerShell script file, please run the following command to enable the PowerShell script to run with out issues: Set-ExecutionPolicy Unrestricted
Question:	**My LDM portal is not updating with device information, or my device is showing "Offline". What should I do?**
Answer:	This typically happens when the Universal Device Client (UDC) has stopped running on the device. To fix: 1. On CMD prompt, run 'services.msc' command 2. When Services application opens, you will see a list of services. Search for 'Universal Device Client'. 3. Check the status column. If status does not show "Running" it needs to be restarted.

4. Highlight the Universal Device Client and right click and select 'Restart' to start the service. This is an automatic service so it will start updating the LDM portal soon after.

Question:	**I see "Information not currently available. Device network still pending" message on my Device Tray. What should I do?**
Answer:	Contact you Org Admin to assign a License to this device. A license may need to be purchased if none are currently available for the organization.
Question:	**I installed the Intel vPro® agent on a device and I want to uninstall it. How can I do this?**
Answer:	Remote uninstallation of the Intel vPro® agent is not currently support. This feature will be added in the next release. For now, removing the agent will require manual interaction with the device.
Question:	**When claiming a device and "Downloading Provisioning Pack", I receive the message: "An error occurred. Please try again". What do I do?**
Answer:	Please wait 10 minutes and try again. If the problem persists, contact Lenovo to investigate further.
Question:	**I cannot see "Settings Restrictions" and/or "App Restrictions" on my Android Device Tray. What should I do?**
Answer:	There are two possibilities for this issue. Please check: 1. Contact your Org Admin to ensure the Android Settings/Application Management has been enabled under Organization Settings. The default setting is enabled, but if it was disabled, this feature will not appear. 2. Device may not have been claimed as 'Device Owner'. Reclaim the device as Device Owner to resolve. If the feature has been enabled in Org Settings AND the device was claimed as Device Owner, please wait 10 minutes and try again, giving LDM time to recognize the device under Device Owner status. If the problem persists, contact Lenovo to investigate further.

9 Reference Documents

Terms & Conditions: Available on LDM portal “Preferences”

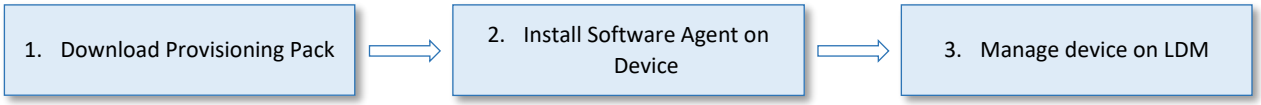
Lenovo Privacy Policy

Lenovo Software as a Service Cloud Agreement

10 LDM Quick Start Guide

Overview

The purpose of this guide is to help you smoothly onboard one or more devices in your organization to the Lenovo Device Manager platform.



Note for New User: To set up a new LDM account, it is mandatory to have a Lenovo ID and get an email invitation registered with Lenovo. Once admin account is set up, the administrator can invite other users within the organization to create accounts based on the roles and permissions granted to them.

You can onboard one or more devices to LDM platform through the LDM→Device Management / Devices page. This process may vary based on the device type and operating system as described below. This guide will provide quick instructions for each category type.

	Current Device Type Categories			
	PCs	**SmartEdge**	**Tablet / Mobile**	**AR / VR**
Examples	Any Lenovo notebook, desktop, workstation, etc	Any Lenovo edge appliance and servers	Any Lenovo tablets or mobile device	Any Lenovo AR or VR appliance
Current Install Options	• Automatically with Provisioning Package download	• Automatically with Provisioning Package download	• Automatically with QR code scan	Currently Unavailable in LDM 2.1 *Will return in LDM 2.2*

The UDC software agent bundle includes:

- setup.cmd
- udc.exe
- udc-provision.json
- Readme file

10.1 Add Windows Devices

10.2 Add Android Devices

10.3 Installing the Intel vPro® Agent

10.4 Requirements

10.5 Troubleshooting

11 App Package – Example PowerShell Script

Custom package creation allows you to package and deploy applications and policies to your devices using the specifications outlined in Section 5.2.2 of this User Guide. For reference, use the following PowerShell script as an example:

```
Param(
   [string]$command="install"
)
$pathToSelf = Split-Path -Parent -Path $PSCommandPath
```

```
$pathToLogFile = "$($env:TEMP)\Lenovo.AppPerformance.Package-$(Get-Date -Format 'yyyy-MM-dd_HH-mm-ss').txt"
$taskName = "Lenovo App Performance Task"
$taskFolder = "Lenovo"
$pathToService = "$($env:ProgramData)\Lenovo\Ldi\Performance"
$pathToServiceParent = Split-Path -Parent $pathToService
$pathToRegistry = "HKLM:\Software\Lenovo\Ldi"
$relativePathToLog = "Lenovo\Ldi"

function Write-Log
{
   Param(
      [string]$logString,
      [switch]$isError
   )

   $log = (Get-Date -Format s).ToString() + ": " + $logString
   Add-Content $pathToLogFile -value $log

   if($isError.IsPresent) {
      Write-Error $log
   }else{
      Write-Host $log
   }
}

function Assert-Elevation()
{
   $isAdminOrSystem = ([Security.Principal.WindowsPrincipal] `
    [Security.Principal.WindowsIdentity]::GetCurrent() `
   ).IsInRole([Security.Principal.WindowsBuiltInRole]::Administrator)

   if(-not ($isAdminOrSystem))
   {
      Write-Log "Error: This script requires elevation" -isError
      exit 1
   }
}
```

```
function Set-FolderSecurity {
    Param(
        [Parameter(Mandatory = $true)]
        [string]$path
    )
    $fullControlEnum = [System.Security.AccessControl.FileSystemRights]::FullControl
    $readExecuteEnum = [System.Security.AccessControl.FileSystemRights]::ReadAndExecute
    $allowEnum = [System.Security.AccessControl.AccessControlType]::Allow
    $inheritanceFlag = [System.Security.AccessControl.InheritanceFlags]::ObjectInherit -bor
[System.Security.AccessControl.InheritanceFlags]::ContainerInherit
    $propagationFlag = [System.Security.AccessControl.PropagationFlags]::None

    if (-not (Test-Path $path -PathType Container)) {
        New-Item -Path $path -ItemType Directory
    }

    $acl = Get-Acl -Path $path

    # takeown
    $adminGroup = New-Object System.Security.Principal.NTAccount("Builtin", "Administrators")
    $acl.SetOwner($adminGroup)

    # disable inheritance from parent folder
    $isProtected = $true
    $preserveInheritance = $false
    $acl.SetAccessRuleProtection($isProtected, $preserveInheritance)

    # set permission for different user and group
    $adminAccessRule = New-Object -TypeName System.Security.AccessControl.FileSystemAccessRule -ArgumentList "BUILTIN\Administrators",
$fullControlEnum, $inheritanceFlag, $propagationFlag, $allowEnum
    $systemAccessRule = New-Object -TypeName System.Security.AccessControl.FileSystemAccessRule -ArgumentList "NT AUTHORITY\SYSTEM",
$fullControlEnum, $inheritanceFlag, $propagationFlag, $allowEnum
    $userAccessRule = New-Object -TypeName System.Security.AccessControl.FileSystemAccessRule -ArgumentList "BUILTIN\Users",
$readExecuteEnum, $inheritanceFlag, $propagationFlag, $allowEnum

    $acl.AddAccessRule($adminAccessRule)
    $acl.AddAccessRule($systemAccessRule)
```

```
    $acl.AddAccessRule($userAccessRule)

    Set-Acl -Path $path -AclObject $acl

    # Grant permission to avoid no enough permission when uninstall
    $acl.SetAccessRuleProtection($false, $true)
    Get-ChildItem $path -Recurse -Force | ForEach-Object { Set-Acl -Path $_.FullName -AclObject $acl }
}

function Install()
{
    Uninstall
    Copy-Service
    Add-ScheduledTask
}

function Copy-Service()
{
    Set-FolderSecurity $pathToServiceParent
    Copy-Item $pathToSelf\bin\ai\ $pathToService -Force -Recurse
    if(-not (Test-Path $pathToService -PathType Container))
    {
        Write-Log "Error: Can not copy service to $pathToService" -isError
        exit 1
    }
}

function Add-ScheduledTask()
{
    $triggerTime = "12:00"
    $taskCommand = Join-Path $pathToService 'Lenovo.AppPerformance.exe'
    $taskParameter = ' '
    $settings = New-ScheduledTaskSettingsSet -DontStopIfGoingOnBatteries
    $principal = New-ScheduledTaskPrincipal -GroupId "BUILTIN\Users"

    $action = New-ScheduledTaskAction -Execute $taskCommand -Argument $taskParameter -WorkingDirectory $pathToService
    $triggers =  @(
        $(&{
```

```
            $dailyTrigger = $(New-ScheduledTaskTrigger -Daily -At $triggerTime)
            $dailyTrigger.StartBoundary = [DateTime]::Parse($dailyTrigger.StartBoundary).ToLocalTime().ToString("s")
            $dailyTrigger
        }),
        $(&{
            $logonTrigger = $(New-ScheduledTaskTrigger -AtLogon)
            $logonTrigger.delay = 'PT15M'
            $logonTrigger
        })
    )

    if(-not (Test-path $taskCommand)) {
        Write-Log "Error: Can not find necessary task target $taskCommand " -isError
        exit 1
    }

    Remove-ScheduledTask
    Register-ScheduledTask -TaskName $taskName -TaskPath $taskFolder -Action $action -Trigger $triggers -Settings $settings -Principal $principal

    if (-not ($(Get-ScheduledTask -TaskName $taskName -ErrorAction SilentlyContinue).TaskName -eq $taskName)) {
        Write-Log "Error: Can not create scheduled task." -isError
        exit 1
    }
}

function Uninstall()
{
    Remove-ScheduledTask
    Remove-RegistryKey
    Remove-LogFile

    Set-FolderSecurity $pathToServiceParent
    if (Test-Path $pathToService)
    {
        Remove-Item $pathToService -Recurse -Force
    }
    if (-not (Test-Path (Join-Path $pathToServiceParent "*")))
    {
```

```
        Remove-Item $pathToServiceParent -Recurse -Force
    }
}

function Remove-ScheduledTask()
{
    if ($(Get-ScheduledTask -TaskName $taskName -ErrorAction SilentlyContinue).TaskName -eq $taskName) {
        Unregister-ScheduledTask -TaskName $taskName -Confirm:$False
    }
}

function Remove-RegistryKey()
{
    if (Test-Path $pathToRegistry)
    {
        Remove-Item $pathToRegistry -Recurse -Force
    }
}

function Get-AppDataFolderForAllUsers()
{
    $folderName = "Local Appdata"

    $userProfileList = Get-ItemProperty "Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\*" -Name
"ProfileImagePath" `
    | Where-Object PsChildName -Match "^S-1-5-21.*" `
    | Select-Object PSChildName, ProfileImagePath

    return $userProfileList | ForEach-Object {
        $userShellFoldersKey = "Registry::HKEY_USERS\" + $_.PSChildName + "\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell
Folders";
        if (Test-Path $userShellFoldersKey) {
            $unexpandedFolderPath = (Get-Item $userShellFoldersKey).GetValue($folderName, `
                [System.String]::Empty, `
                [Microsoft.Win32.RegistryValueOptions]::DoNotExpandEnvironmentNames)
            return $unexpandedFolderPath -replace "%USERPROFILE%", $_.ProfileImagePath
        }
    }
```

```
}

function Remove-LogFile()
{
   Get-AppDataFolderForAllUsers | ForEach-Object {
      $userLogFolder = Join-Path $_ $relativePathToLog
      if (Test-Path $userLogFolder) {
         Remove-Item $userLogFolder -Recurse -Force
      }
   }
}

if($command -eq "install")
{
   Assert-Elevation
   Install
}

if($command -eq "uninstall")
{
   Assert-Elevation
   Uninstall
}
```