


Radarr Multiple Instances

Requirements and how to install multiple instances of Radarr

Multiple Instances

It is possible to run multiple instances of Radarr. This is typically done when one wants a 4K and 1080p copy of a movie. Note that you can (and probably should) [review TRaSH's guide and configure Radarr to use a second Radarr as a list](#) . This is helpful if you wish to keep both in sync.

Prerequisites

The following requirements should be noted:

- ▶ If non-docker, the same binaries (windows: program files) should be used
- ▶ If non-docker, all instances *must* have a `-data=` or `/data=` argument passed
- ▶ If non-docker, different ports must be used
- ▶ If non-docker, disable automatic updates on all but 1 instance.
- ▶ If docker, different external (host) ports must be used
- ▶ If docker different host mounts for `/config` must be used
- ▶ Different download client categories must be used
- ▶ Different root folders must be used.

Setting Up Multiple Instances

[Windows Multiple Instances](#)

[Linux Multiple Instances](#)

[Docker Multiple Instances](#)

Windows Multiple Instances

This guide will show you how to run multiple instances of Radarr on Windows using only one base installation. This guide was put together using Windows 10; if you are using a previous version of Windows (7, 8, etc.) you may


need to adjust some things. This guide also assumes that you have installed Radarr to the default directory, and your second instance of Radarr will be called Radarr-4K. Feel free to change things to fit your own installations, though.



Note: You should run Radarr as **either** [a service](#) or as a [tray app](#). Running both an app and a service is unnecessary and likely to cause issues.

Service (Windows)

Prerequisites (Service)

- ▶ [You must have Radarr already installed](#)
- ▶ You must have [NSSM \(Non-Sucking Service Manager\)](#)  installed. To install, download the latest release (2.24 at the time of writing) and copy either the 32-bit or 64-bit nssm.exe file to C:/windows/system32.
 - ▶ If you aren't sure if you have a 32-bit or 64-bit system, check Settings => System => About => System type.

Configuring Radarr Service

1. Open a Command Prompt administrator window. (To run as an admin, right click on the Command Prompt icon and choose "Run as administrator.")

2. If Radarr is running, stop the service by running `nssm stop Radarr` in Command Prompt.

3. Now we have to edit the existing Radarr instance to explicitly point to its data directory. The default command is as follows:

```
sc config Radarr binpath= "C:\ProgramData\Radarr\bin\Radarr.exe - data=C:\ProgramData\Radarr"
```

This command tells the original instance of Radarr to explicitly use

`C:\ProgramData\Radarr` for its data directory. If you didn't use the default Radarr install, or if your data folder is somewhere else, you may have to change your paths here.

Creating Radarr-4K Service

1. Create a new folder where you'd like Radarr-4K to live. Most use a similar place such as

`C:\ProgramData\Radarr-4K`

2. Back in Command Prompt, create the new Radarr-4K service using `nssm install Radarr-4K`. A popup window will open where you can type your parameters for the new instance. For this example, we will use the following:

▶ Path: `C:\ProgramData\Radarr\bin\Radarr.exe`

▶ Startup directory: `C:\ProgramData\Radarr\bin`

- ▶ Arguments: `-data=C:\ProgramData\Radarr-4K`
- ▶ Exit Actions Tab
 - ▶ Restart: Restart Application
 - ▶ Delay: 120000 ms
(2 minutes, can be longer if update fails to complete in time)



Note that **Arguments** points to the *new* folder created in step 1. This is crucial, as it keeps all the data files from both instances in separate locations.

1. Click *Install service*. The window should close and the service will now be available to run.
2. Continue to [Configuring Radarr-4k](#)

Tray App (Windows)

Prerequisites (Tray App)

- ▶ [You must have Radarr already installed](#)
- ▶ Radarr's shortcut must be configured with a `/data=` argument in the 'target' field to allow multiple instances
- ▶ Navigate to the Startup Folder for the current user `%appdata%\Microsoft\Windows\Start Menu\Programs\Startup` and edit the existing shortcut if needed.

Creating Radarr-4K Tray App

- ▶ Create a new folder for Radarr-4K's configuration files. Most use a similar place such as `C:\ProgramData\Radarr-4K`
- ▶ Right click and Create New Shortcut
- ▶ Path: `C:\ProgramData\Radarr\bin\Radarr.exe /data=C:\ProgramData\Radarr-4K`
- ▶ Give the shortcut a unique name such as `Radarr-4K` and finish the wizard.
- ▶ Double click the new shortcut to run and test.
- ▶ Continue to [Configuring Radarr-4k](#)

Configuring Radarr-4k

- ▶ Regardless of if you used the Service Method or the Tray App: Stop both services and both Apps
- ▶ Start Radarr-4k (Service or Tray App)
- ▶ Open up Radarr-4k and Navigate within the app to [Settings => General => Host](#)

- ▶ Change **Port Number** from **7878** to a different port e.g. **7879** so Radarr and Radarr4k do not conflict
- ▶ You should now be able to start both apps
- ▶ Continue to [Dealing with Updates](#)

Dealing with Updates



▶ Disable automatic updates in one of your instances

- ▶ In config.xml change update branch to **<Branch>nonexistent</Branch>**
- ▶ If one Radarr instance is updated, both instances will shutdown and only the updated one will start again. To fix this, you will have to manually start the other instance, or you may want to look into using the below powershell script to address the problem.



Configuring the [NSSM Exit Action](#) correctly should allow Radarr to update and restart multiple instances with no additional scripts.

If the restart delay is not configured by default it will restart the instance immediately.

This can prevent updates from being applied and can result in the following error **Radarr was restarted prematurely by external process.**

Windows Port Checker and Restarter PowerShell Script

- ▶ When you use two Radarr instances and one of it is updating, it will kill all instances. Only the one which is updating will come back online.
 - ▶ The below powershell script should be configured as a scheduled task.
 - ▶ It checks the ports and if one is not online, it will (re-)start the scheduled task to launch Radarr.
1. Create a new File and name it RadarrInstancesChecker.ps1 with the below code.
 2. Edit the script with your actual service names, IP, and ports. *If you are running in Tray mode, you must create Scheduled tasks to start each Radarr instance and use those Task names in the script below.*
 3. [Create a scheduled task](#) to run the script on a repeating schedule.
- ▶ Security Options: Enable **Run with highest privileges**
 - ▶ Otherwise the script will be unable to manipulate services
 - ▶ Trigger: **On Launch**
 - ▶ Repeat task every: **5** or **10** minutes
 - ▶ Action: **Start a Program**
 - ▶ Program/script: **powershell**
 - ▶ Argument: **-File D:\RadarrInstancesChecker.ps1**

- Be sure to use the full path to your script's location

```
#####
### RadarrInstancesChecker.ps1
#####
### Keeps multiple Radarr Instances up by checking the port
### Please use Radarr's Discord or Reddit for support!
### https://wiki.servarr.com/radarr/installation#windows-multi
#####
### Version: 1.1
### Updated: 2020-10-22
### Author: reloxx13
#####

### SET YOUR CONFIGURATION HERE ###
# Set your host ip and port correctly and use your service or scheduledtask names

# (string) The type how Radarr is starting
# "Service" (default) Service process is used
# "ScheduledTask" Task Scheduler is used
$startType = 'Service'

# (bool) Writes the log to C:\Users\YOURUSERNAME\log.txt when enabled
# $false (default)
# $true
$logToFile = $false

$instances = @(
    [pscustomobject]@{ # Instance 1
        Name = 'Radarr-V3'; # (string) Service or Task name (default: Radarr-V3)
        IP = '192.168.178.12'; # (string) Server IP where Radarr runs (default:
        Port = '7873'; # (string) Server Port where Radarr runs (default: 7873)
    }
    [pscustomobject]@{ # Instance 2
        Name = 'Radarr-4K'; # (string) Service or Task name (default: Radarr-4K)
        IP = '192.168.178.12'; # (string) Server IP where Radarr runs (default:
        Port = '7874'; # (string) Server Port where Radarr runs (default: 7874)
    }
    # If needed you can add more instances here... by uncommenting out the below
    # [pscustomobject]@{ # Instance 3
    # Name='Radarr-3D'; # (string) Service or Task name (default: Radarr-3D)
    # IP='192.168.178.12'; # (string) Server IP where Radarr runs (default: 192.1
    # Port='7875'; # (string) Server Port where Radarr runs (default: 787
    # }
)
```

```

)

### DONT CHANGE ANYTHING BELOW THIS LINE ###

###
# This function will write to a log file or in console output
###
function Write-Log
{
    #Will write to C:\Users\YOURUSERNAME\log.txt

    Param(
        $Message,
        $Path = "$env:USERPROFILE\log.txt"
    )

    function TS { Get-Date -Format 'hh:mm:ss' }

    #Console output
    Write-Output "[$(TS)]$Message"

    #File Output
    if ($logToFile)
    {
        "[$(TS)]$Message" | Tee-Object -FilePath $Path -Append | Write-Verbose
    }
}

Write-Log 'START ====='

$instances | ForEach-Object {
    Write-Log "Check $($_.Name) $($_.IP):$($_.Port)"

    $PortOpen = ( Test-NetConnection $_.IP -Port $_.Port -WarningAction SilentlyContinue)

    if (!$PortOpen)
    {
        Write-Log "Port $($_.Port) is closed, restart $($startType) $($_.Name)!"

        if ($startType -eq 'Service')
        {
            Get-Service -Name $_.Name | Stop-Service
            Get-Service -Name $_.Name | Start-Service
        }
    }
}

```

```

94         elseif ($startType -eq 'ScheduledTask')
95         {
96             Get-ScheduledTask -TaskName $_.Name | Stop-ScheduledTask
97             Get-ScheduledTask -TaskName $_.Name | Start-ScheduledTask
98         }
99         else
100        {
101            Write-Log '[ERROR] STARTTYPE UNKNOWN! USE Service or ScheduledTask !'
102        }
103    }
104    else
105    {
106        Write-Log "Port $($_.Port) is open!"
107    }
108 }
109
110 Write-Log 'END ====='

```

Linux Multiple Instances

► [Swizzin Users](#)

► Non-Swizzin Users

- Ensure your first instance has the `-data=` argument passed.
- Temporarily stop your first instance, so you can change the second instance's port `systemctl stop radarr`
- Disable automatic updates on one of your Radarr Instances`



Below is an example script to create a Radarr4K instance. The below systemd creation script will use a data directory of `/var/lib/radarr4k/`. Ensure the directory exists or modify it as needed.

```

1  cat << EOF | sudo tee /etc/systemd/system/radarr4k.service > /dev/null
2  [Unit]
3  Description=Radarr4k Daemon
4  After=syslog.target network.target
5  [Service]
6  User=radarr
7  Group=media
8  Type=simple
9
10 ExecStart=/opt/Radarr/Radarr -nobrowser -data=/var/lib/radarr4k/
11 TimeoutStopSec=20

```

```
12 | KillMode=process
13 | Restart=on-failure
14 | [Install]
15 | WantedBy=multi-user.target
16 | EOF
```

- Reload systemd:

```
1 | sudo systemctl -q daemon-reload
```

- Enable the Radarr4k service:

```
1 | sudo systemctl enable --now -q radarr4k
```

Docker Multiple Instances

- Simply spin up a second Docker container with a different name, ensuring the [above requirements](#) are met.