

# How to script Nipper Studio

## Before we start

This document makes use of the command line version of Nipper Studio to perform an audit on Microsoft Windows, GNU/Linux or Apple Mac. It will assume that you can execute the command line version by entering the command “nipper”.



Nipper Studio 2.5.2 installs the command line tool by default and adds Nipper Studio to the global path to allow users to launch Nipper Studio from any directory and run audits easier and more efficient. This feature will be present in all future versions of Nipper Studio.

With all version of Nipper Studio up to and including 2.5.1.4644, the default installation options for Nipper Studio do not include the command line tools. If “nipper.exe” is not included in your Nipper Studio directory, then you will have to re-run setup and include it on the installation options screen.

When installing, the Nipper Studio directory is not automatically added to the Windows search path. To be able to execute “nipper” you will need to open a command prompt in the directory where you installed the software, or use the full path when you run it. E.g. “C:\Program Files\NipperStudio\nipper” instead of just “nipper”.

Throughout this document we will detail command line options and give examples of how they are used. If the commands include any of the conventions detailed in the table below, then the convention and its contents should be replaced entirely. The conventions used in this document are:

Convention	Description
[ ]	Describes optional command text – it is not required.
< >	Describes an option that you must choose.
	Used to divide a list of options that could be used – it will always appear within one of the previous conventions.

Nipper Studio is a powerful auditing tool with a huge number of command line options. The command line tool includes a built-in help facility. To access this, just enter the following command:

```
nipper
```

The help has been split into topics such as “Report”, “Devices” and “Save”. To get help on any given topic, enter the command:

```
nipper --help=<topic>
```

For example, you can get help on setting profiles by using the following command:

```
nipper --help=profile
```

## The basics – auditing using the command line

You can perform an audit of a device configuration that has been saved to a local file using the following command:

```
nipper --input=<configuration file> --output=<report file>
```

For example, to audit the demo configuration file “cisco-router.txt” from the “demo-files” directory and save the report to my Windows desktop, I could enter:

```
nipper --input=demo-files\cisco-router.txt --output=c:\users\testuser\Desktop\test.html
```

To audit a device over the network (remote), such as a Cisco router, you could use a command such as the following:

```
nipper --remote-device=<address> --ios-router --protocol=<ssh | https> --  
username=<user> --password=<password> --priv-password=<enable password> --  
output=<report file>
```

If you want to audit multiple devices in the same report, you can just add other configurations as an input for Nipper Studio:

```
nipper --input=demo-files\cisco-router.txt --input=demo-files\3com5500.txt --  
output=c:\users\testuser\Desktop\test.html
```

For auditing multiple remote devices, you can stack the commands such as:

```
nipper --remote-device=<address> --ios-router --protocol=<ssh | https> --  
username=<user> --password=<password> --priv-password=<enable password> --remote-  
device=<address> --juniper-srx --protocol=<ssh | https> --username=<user> --  
password=<password> --output=<report file>
```

To audit a directory containing configuration files from lots of different devices you could use a command such as:

```
nipper --input=<configuration directory> --output=<report file>
```

All those examples assume that we are using the Nipper Studio settings that are currently configured and accepting any defaults. You can configure other settings by adding them to the command too, for example:

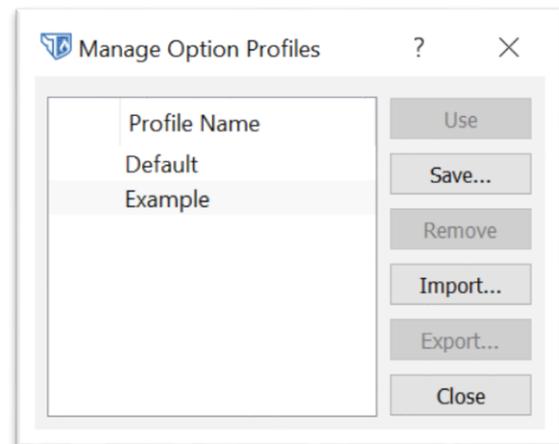
```
nipper --input=<configuration file> --output=<report file> --company-name=Titania
```

## Using settings profiles

The Nipper Studio command line is a powerful tool with lots of options. Profiles make it easy for you to use the same settings each time and you can have multiple profiles for each different type of audit that you want to perform.

All settings and profiles are shared between both the graphical interface and the command line. So you may find it easier to browse the Nipper Studio settings using your mouse in the graphical interface. Once you have set the options for a particular profile, select the “Manage Option Profiles” from the “Tools” menu, click “Save” and give the profile a name.

In the screenshot to the right, we have called our profile “Example”. The “Default” profile represents the default settings for Nipper Studio and can be used to revert back to the original factory default settings.



To change settings using the command line, you can enter one or many settings at a time on the command line:

```
nipper --company-name=Titania
```

All the settings are saved, so they are remembered between each time you run Nipper Studio. This functionality can be changed by modifying the auto save setting:

```
nipper --autosave=<on | off>
```

For the purpose of this guide though, it is helpful for auto save to be enabled (the default setting). To save the current settings as a new profile, you can use the following command:

```
nipper --save-profile=<name>
```

Existing profiles can be listed using the following command:

```
nipper --show-profiles
```

If a current profile is being used, it will be shown with a \* next to the name.

Now when you run an audit you can specify the name of the profile to use at the start and all the options set for that profile will be used.

```
nipper -use-profile=<profile name> --input=<configuration file> --report=<report file>
```

## Scheduling an audit

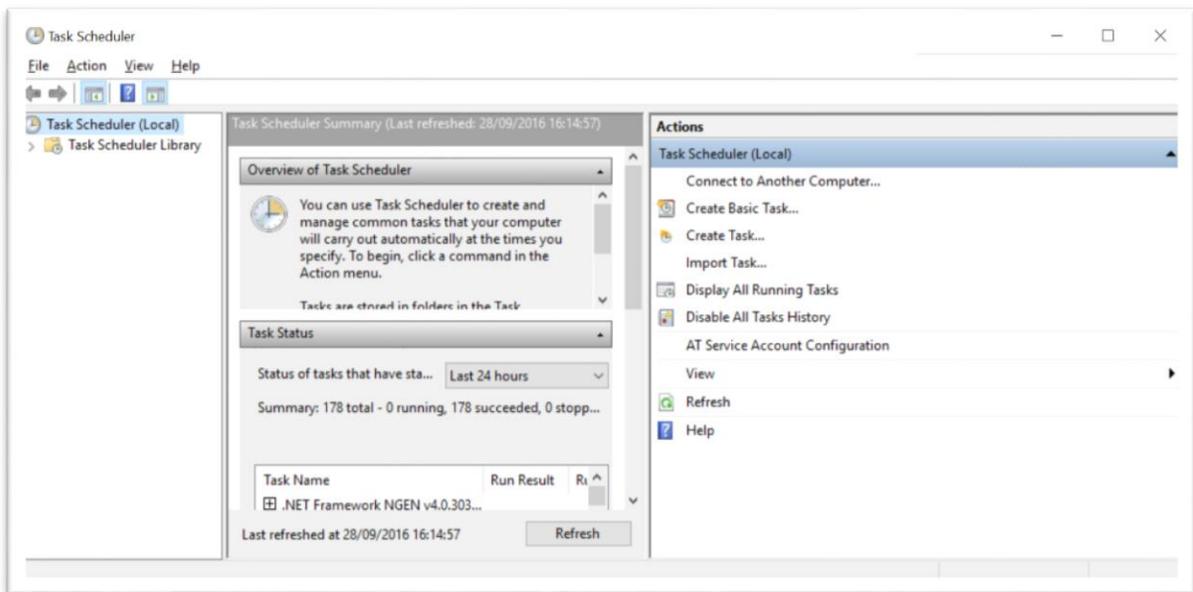
How you can schedule a Nipper Studio audit depends on which operating system that you use for Nipper Studio. This guide will cover using Task Scheduler (Windows) and Cron (GNU/Linux) for creating and scheduling an audit task.

As an example, we will focus on creating a scheduled task to perform a daily audit of all the configurations stored in a single directory. We will also use a profile to ensure that the same settings are used for each audit.



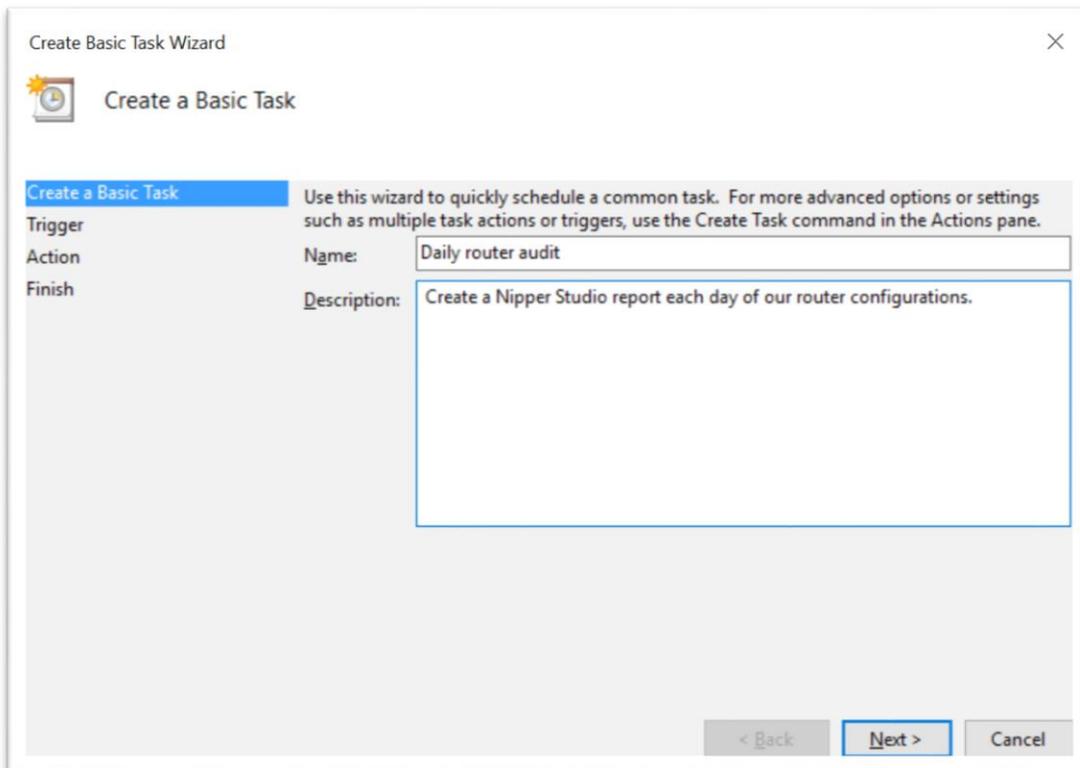
Microsoft Windows

1. Start the Windows Task Scheduler (on Windows 10 you can do this by clicking on the Start button and then entering “task scheduler” and clicking on “Task Scheduler”).



2. Click on the “Create Basic Task...” action.

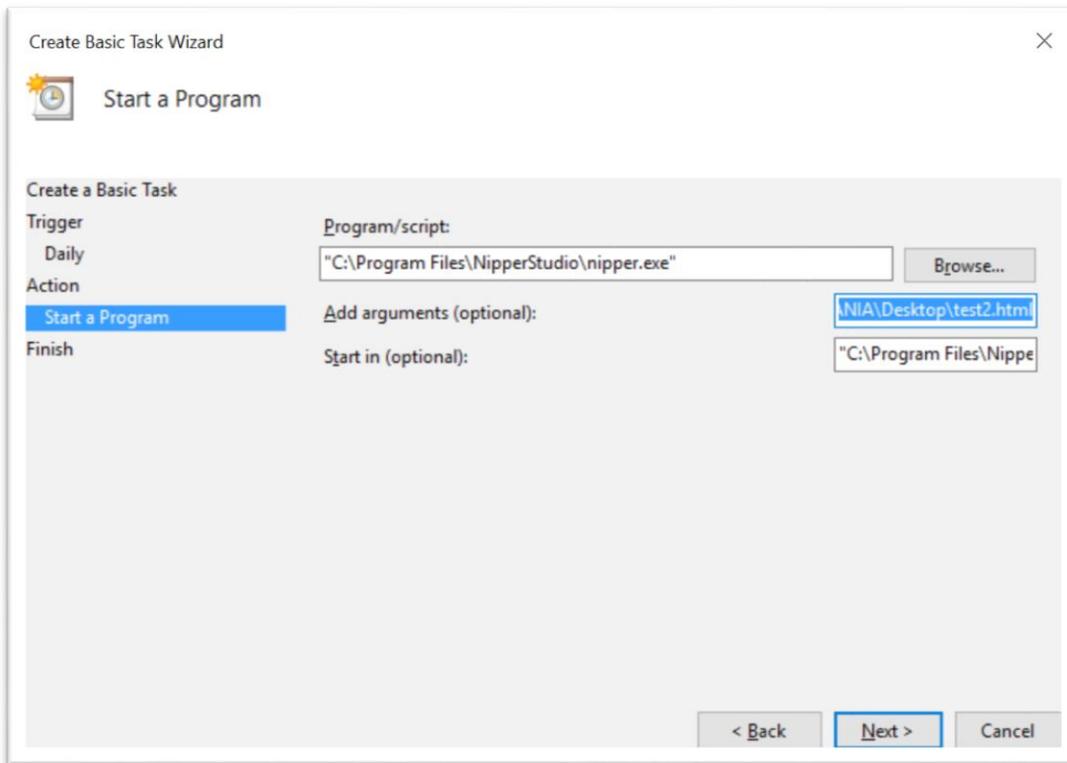
3. Enter a name and description for the task, then click on the “Next” button.



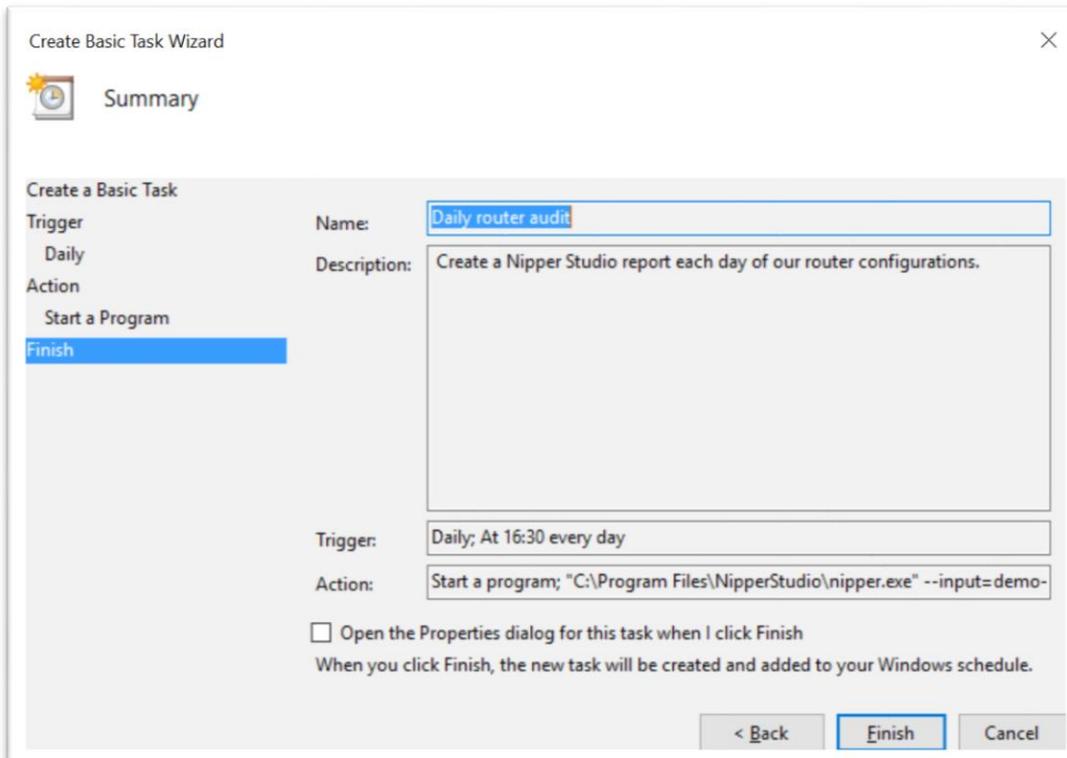
The screenshot shows a 'Create Basic Task Wizard' dialog box. The title bar reads 'Create Basic Task Wizard' with a close button (X) in the top right corner. Below the title bar is a header area with a clock icon and the text 'Create a Basic Task'. The main area is divided into a left sidebar and a right content area. The sidebar has four items: 'Create a Basic Task' (highlighted in blue), 'Trigger', 'Action', and 'Finish'. The right content area contains a grey instruction box: 'Use this wizard to quickly schedule a common task. For more advanced options or settings such as multiple task actions or triggers, use the Create Task command in the Actions pane.' Below this are two input fields: 'Name:' with the text 'Daily router audit' and 'Description:' with the text 'Create a Nipper Studio report each day of our router configurations.'. At the bottom right, there are three buttons: '< Back' (disabled), 'Next >' (active/highlighted), and 'Cancel' (disabled).

4. Select “Daily” for the trigger, then click on the “Next” button.
5. Select a start time for the task, then click on the “Next” button.
6. Select “Start a program” from the actions, then click on the “Next” button.

7. Select the "nipper.exe" program in the "program/script" box. Then add the arguments "--input=<configuration directory> --report=<report file>". Also add the nipper.exe program file directory to the "Start in (optional)" box – DO NOT USE QUOTES. Then click on the "Next" button.



8. Review the summary and click "Finish"



The audit task should now be scheduled and run on daily at the specified time.

It is worth noting that there are a number of things on Windows that may prevent Task Scheduler from allowing a task to successfully run. The Task Scheduler application shows event viewer entries that are relevant and can help to diagnose any issues. Below are a few common causes for scheduled tasks not to run.

- Your user needs permission to schedule tasks.
- Your Windows system may report that the user is not logged in. In Task Scheduler, modify the properties of the task and set it to “Run whether user is logon on or not”. You will need to enter the logon password for your user at the logon prompt.
- You should not have any quotes in the “Start in” directory. If this is not possible, then you can always create a windows batch file to execute the audit job, rather than directly executing Nipper Studio.

To create a Windows batch file:

1. Open notepad.
2. Enter one command on each line, as you would type it on the command line.
3. Once you have added all the commands, save the file with a “.bat” file extension. For example, “audit.bat”.



## GNU/Linux

To schedule an audit on GNU/Linux systems, such as Red Hat or Ubuntu, this guide will use the Cron scheduler system.

1. Save a copy of the current Cron table by using the following command to save it to a file called "crontab.txt":

```
crontab -l > crontab.txt
```

2. Open the "crontab.txt" file using your favorite text editor. There may or may not already been entries in the schedule.
3. The file is expected to be in a specific format, with comment lines beginning with a "#" character. The format covers six individual elements, each separated by a space. They are minute (0-59), hour (0-23), day-of-month (1-31), month (1-12), day-of-week (0-6 starting with Sunday) and the command. A "\*" is used wherever a value should be "any".
4. Add the following line to the file to schedule a Nipper Studio audit at 11:30pm every day:

```
30 23 * * * nipper --input=<configuration directory> --report=<report file>
```

5. Save the Cron table and exit your text editor.
6. Make Cron use the new table using the following command:

```
crontab crontab.txt
```

Now you should have an audit job scheduled on your Linux system.

Just like with Windows Scheduler, there may be something that prevents a Cron job from being run. In those cases, the following may help:

- Check the log files (usually in /var/log/) to see if there any events that could indicate the reason for the failure.
- Check in "/etc/cron.deny" and "/etc/cron.allow" to see if the user has been denied or given permission to schedule jobs.
- Check to see if the system has the correct time, it may be in a different timezone.

You can run the batch file by clicking on it in Windows Explorer or from the command line. When you are happy that everything is working as expected, follow the task scheduler steps to schedule the audit.