

# Remote.it Bulk Scripting

Remote.it allows you to easily upload and run scripts across multiple remote devices using the bulk scripts feature. This can be done in three easy steps:

1. Create your script
2. Upload your script
3. Execute your script

In the following examples, we will be using the **shell scripting** interface. However, **you may send any valid script using Remote.it, as long as it's supported by the device and is executable**. In our examples, the target system will be a Raspbian based Raspberry Pi.

## Creating a script

Lets looks at an example of a script called "hello-remote-it.sh" that is created for use with Remote.it. Running this script will display "Hello Remote.it!" in column (Status) A of the device list in the Remote.it portal:

### Example 1: hello-remote-it.sh

```
1. #!/bin/sh
2. # hello-remote-it.sh
3.
4. TOOL_DIR="/usr/bin"
5. NOTIFIER=connectd_task_notify\
6.
7. ${TOOL_DIR}/${NOTIFIER} a $1 $2 "Hello Remote.it!"
8.
9. ${TOOL_DIR}/${NOTIFIER} 1 $1 $2 "Job complete"
```

While this may seem confusing at first, this script is actually quite simple! The Remote.it installer provides required utilities that allow you to interface with the Remote.it portal. In this case, will be using **connectd\_task\_notify**, a notification plugin that sends data back to the Remote.it portal. The structure is as follows:

```
connectd_task_notify.sh [cmd] [taskid] [api] [status]
```

**Cmd** : Tells task\_notify what to update. **Input:** 0-2 sends an update status to the job server, and a-e (alt: 3-7) represent the different columns (Statuses) of the device list. The exact values translate to the following:

0 : update status  
1 : completed status  
2 : failed status  
3 - 7 : columns a-e

**TaskID** : The ID of the particular job.

**API** : The API for the particular job.

The taskID and API are sent to your script by the connectd\_schannel command execution daemon. There is nothing for you to do with these other than to make sure they are sent correctly back to connectd\_task\_notify. **NOT COMPLETE**

**Status** : The message you want to be sent back to Remote.it. With options a-e (alt: 3-7), the message will be sent to its corresponding column. With cmd options 0-2, the message will be sent to the Job Status page as a description.

On line 7 of the above example we are returning the values as such:

```
connectd_task_notify.sh [cmd = a] [taskid = $1] [api = $2] [status = "Hello  
Remote.it!"]
```

On line 9, connectd\_task\_notify is also being called at the end of the script:

```
${TOOL_DIR}/${NOTIFIER} 1 $1 $2 "Job complete"
```

**IMPORTANT** : You **must** include a call to connectd\_task\_notify, passing in 1 as the argument for [cmd], to tell the Remote.it portal that the script has now completed successfully, in which it will then change the status in the job list from "running" to "done". Failing to do so will result in the "running" status to appear infinitely until the job is cancelled.

You might have also notice the two variables being declared on lines 4 and 5 of the above example:

```
TOOL_DIR="/usr/bin"
NOTIFIER=connectd_task_notify\
```

**TOOL\_DIR** : Represents the path to the Remote.it tools and utilities, which includes connectd\_task\_notify.

**NOTIFIER** : Represents the actual connectd\_task\_notify tool

While defining the pathways as variables is not required, it allows for quick and easy access to the desired tools, and is useful for scripts that frequently use certain tools and/or paths.

### A Template Structure for Bulk Scripts:

```
1. #!/bin/sh
2.
3. #Store directory paths.
4. TOOL_DIR="/usr/bin"
5. NOTIFIER=connectd_task_notify\
6.
7. #Your script here
8.
9. #Pass "complete" status to Remote.it portal
10. ${TOOL_DIR}/${NOTIFIER} 1 $1 $2 "Job complete"
```

**IMPORTANT** : When saving your script, you must to make sure the line endings/breaks are set to match that of your target device's **operating system**. The default line endings are either:

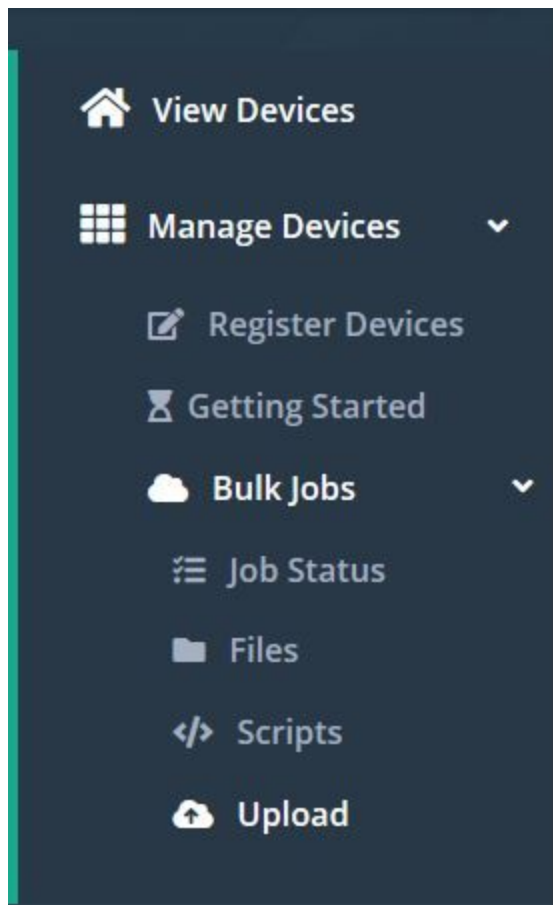
**LF** : Linux and MacOS

**CRLF** : Windows

If you are creating a script on a Windows system, to be run on a Raspberry Pi, then you need to make sure the line endings are set to LF, instead of the default CRLF, which can be done inside of a code editor.

# Uploading a script

A script can be uploaded by logging into Remote.it, navigating over to the side menu, and selecting Manage Devices > Bulk Jobs > Upload.



Once in the Storage Upload menu, select the file you wish to upload, and set the file type to “Executable script or program”. Then press “Upload”.

## Storage Upload

Select files for upload then press Upload button.

Browse...

hello-remote-it.sh

Choose File Type:

☐ General purpose content


☒ Executable script or program


Cancel

Upload

Once the script has upload successfully, it should now be visible in the Script Storage menu (Side menu: Manage Devices > Bulk Jobs > Scripts).

Script Storage ×

Filter 

Name	Size	Created	↓ ↑ Actions
<a href="#">hello-remote-it.sh</a>	161 B	2018-07-27 00:58:21	

Your script is now ready to be executed!

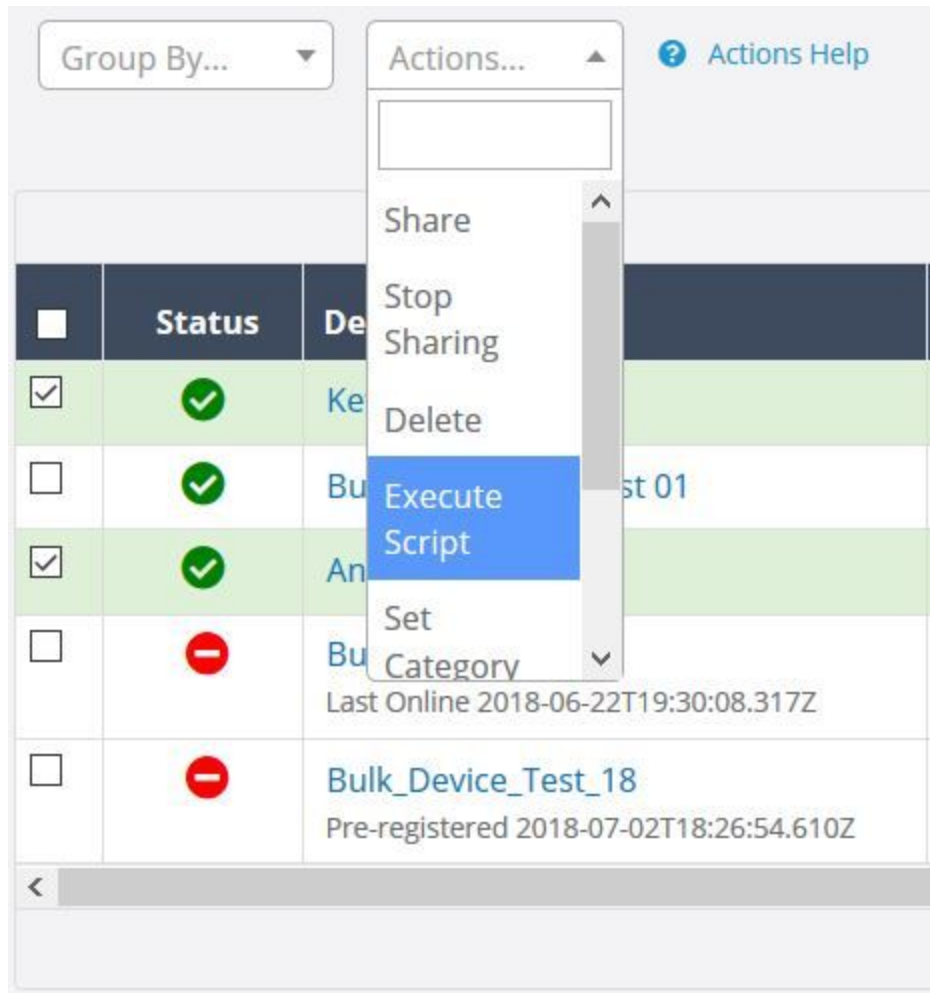
## Executing a script

To execute a script, log into Remote.it and navigate to View Devices, which you can access in the side menu. It is also the homepage of the portal.

Select the device(s) you wish to execute the script on by checking the boxes to the far left, next to the Status column. Doing so should highlight the row green.

<input type="checkbox"/>	Status	Device Name
<input checked="" type="checkbox"/>	✓	Kevins-Pi
<input type="checkbox"/>	✓	Bulk-registration-test 01
<input checked="" type="checkbox"/>	✓	Andrew
<input type="checkbox"/>	✗	Bulk_Device_Test Last Online 2018-06-22T19:30:08.317Z
<input type="checkbox"/>	✗	Bulk_Device_Test_18 Pre-registered 2018-07-02T18:26:54.610Z
<		

Navigate to the Actions menu above the table, and select “Execute Script” from the drop down menu. This will take you to the Bulk Execution menu.



In the Bulk Execution menu, select the file you wish to execute and select “Next”.



# Bulk Execution



The Wizard will guide you through the steps to execute a script.

Select the script first to determine remaining steps

cc-set-bulk-id.sh  
cc-set-hwid-rgky.sh  
**hello-remote-it.sh**  
install-app.sh  
remot3it-demo-script.sh  
run-forever-server.sh

Cancel

Next

Check the box that reads "Check to allow all devices to update the status columns during Job execution.", and hit "Submit".

# Bulk Execution



The Wizard will guide you through the steps to execute a script.

## Final Step: Confirm Bulk Job submission

☒ Check to allow all devices to update the status columns during Job execution.

Pressing the **Submit** button will create the bulk job and queue it for execution.

Cancel

Submit

Once submitted, navigate to the Job Status menu by selecting Manage Devices > Bulk Jobs > Job Status in the side menu.





In the Job Status menu, you should see your newly uploaded script with a status of “new”.

## Job Status

Script	Status
hello-remote-it.sh	new

This should then change from “new” to “running”, and from “running” to “done” if the script ran successfully.

After running the script, you will be able to see that Status A now shows “Hello Remote.it!”

	Status	Device Name	Share 	HWID	Internal IP	External IP	Status A
<input type="checkbox"/>		<a href="#">Kevins-Pi</a>		b8:27:eb:4d:61:e5-n	192.168.42.76	142.254.21.68	Hello Remote.it!

**IMPORTANT:** A script may take up to 3 minutes to process from start to finish, and there may be short delay from when the script actually completes, to when the status is updated on the Remote.it portal. It is also common for the rows in the device list update before the job status shows that it is updated.