Batch Scheduling Process on Node-RED Integration Guide

Introduction

In this comprehensive guide, we will guide you through the process of setting up a batch processing scheduler using Node-RED. This guide will provide you with step-by-step instructions for installing Node-RED alongside some related Node-RED palettes required in this integration. We will demonstrate the integration, showing you how to automate your KNIME workflows.

Please note that this guide is based on the Node-RED version **3.1.3**. Future versions may have different features or requirements, so some steps in this guide might not be applicable.

Prerequisites

For equipments, you will need:

1. Any Windows PC/laptop
2. Network connection

Node-RED Installation

If you already have Node-RED installed, you can skip this step.

The following are instructions on how to install Node-RED on a Windows based operating system. All required download links can be found on the official Node-RED and NodeJS websites.

Node.js

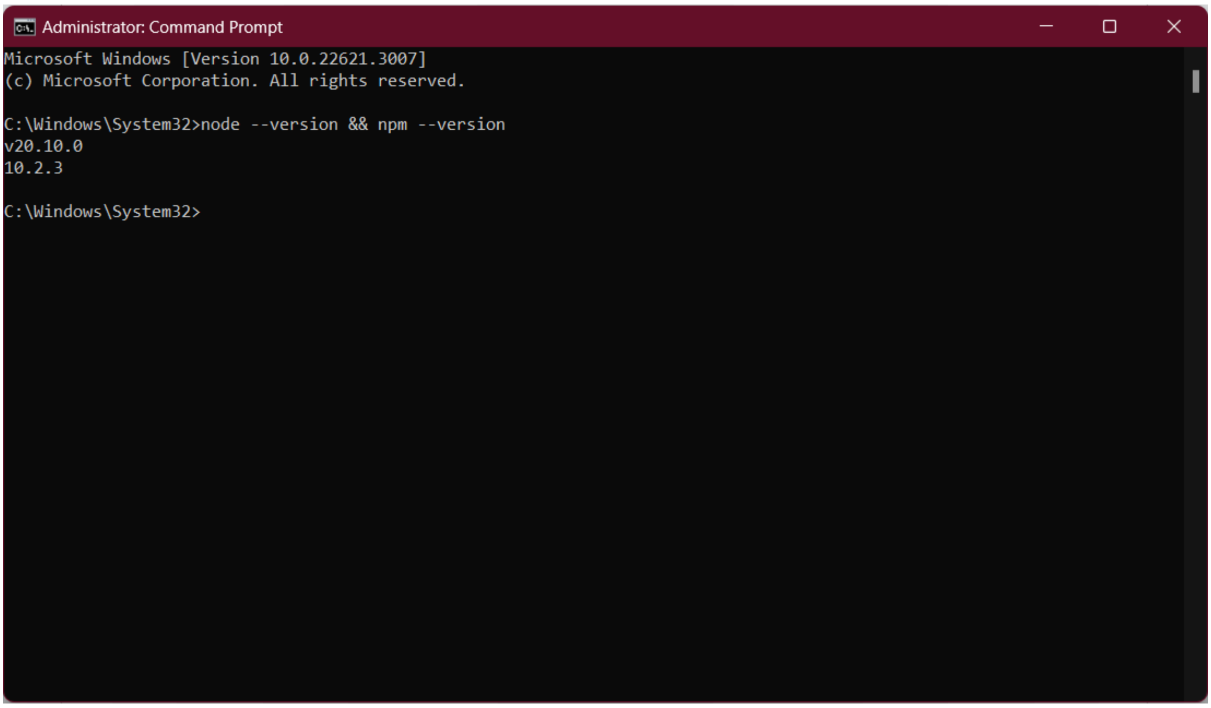
To install Node-RED locally you will need a supported version of Node.js. Node.js is a JavaScript based programming language and must be installed before Node-RED. Node.js can be downloaded from the official Node.js website.

Once the installation is complete, open up a Windows command prompt (CMD), you can open a command prompt by using the search box in your windows taskbar and type cmd and run the application as administrator then once the command prompt has opened, input the following:

```
node --version && npm --version
```

This command checks if both Node.js and NPM are correctly installed, and it will be expected to return a version number.

The output should look something like this:



```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>node --version && npm --version
v20.10.0
10.2.3

C:\Windows\System32>
```

NPM is a package manager for the JavaScript programming language, this is needed for installing additional Node-RED packages which we will need later. NPM should be installed by default when installing Node.js.

Install Node-RED

After installing Node.js, we can begin to install Node-RED, navigate to CMD and input the following command:

```
npm install -g --unsafe-perm node-red
```

That command will install Node-RED as a global module along with its dependencies.

Please note if you are having trouble with this please contact your System Administrator. Once this command is successfully run Node-RED will begin to install and this can take some time to complete. Once the installation is complete, the command command output should look something like:

```
+ node-red@1.1.0
added 332 packages from 341 contributors in 18.494s
found 0 vulnerabilities
```

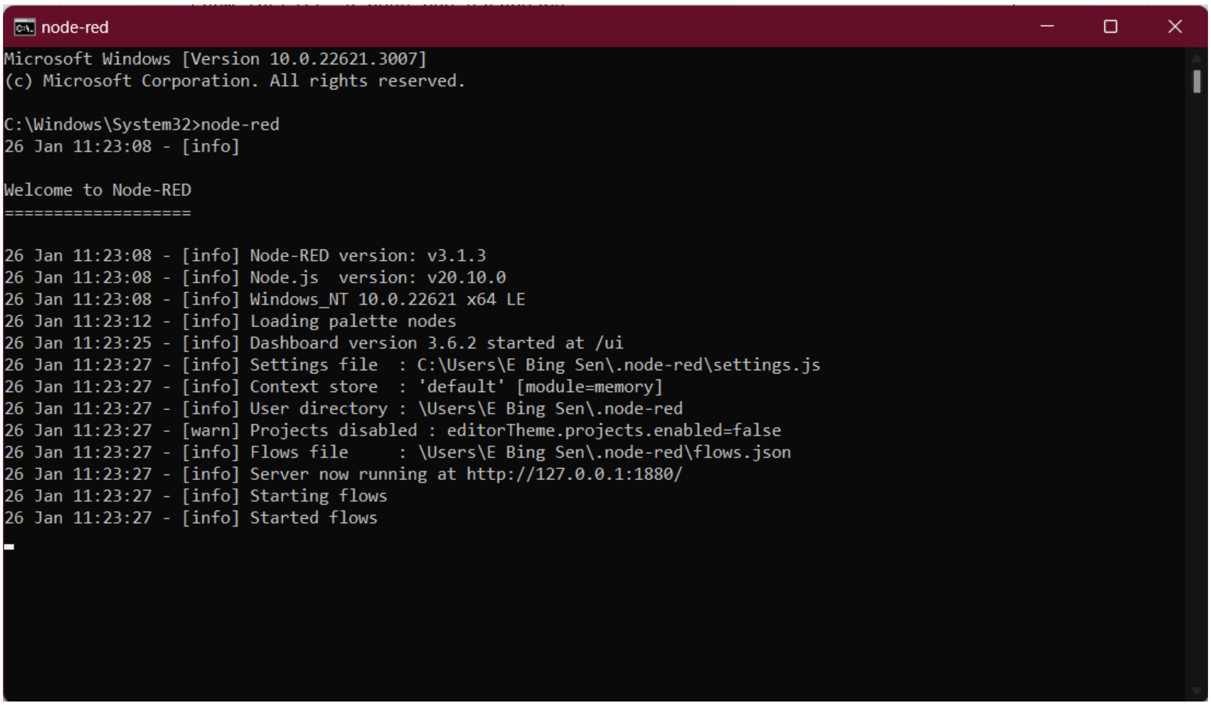
Run Node-RED

If you already know how to run Node-RED, you can skip this step.

You can begin to run your Node-RED web server and install additional packages within Node-RED. You can first do this by navigating to a command prompt and typing in the following command:

```
node-red
```

As we installed Node-RED as a global NPM module it adds the command node-red to your system path which you can then input into a command prompt to start Node-RED:



```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>node-red
26 Jan 11:23:08 - [info]

Welcome to Node-RED
===================

26 Jan 11:23:08 - [info] Node-RED version: v3.1.3
26 Jan 11:23:08 - [info] Node.js  version: v20.10.0
26 Jan 11:23:08 - [info] Windows_NT 10.0.22621 x64 LE
26 Jan 11:23:12 - [info] Loading palette nodes
26 Jan 11:23:25 - [info] Dashboard version 3.6.2 started at /ui
26 Jan 11:23:27 - [info] Settings file  : C:\Users\E Bing Sen\.node-red\settings.js
26 Jan 11:23:27 - [info] Context store  : 'default' [module=memory]
26 Jan 11:23:27 - [info] User directory : \Users\E Bing Sen\.node-red
26 Jan 11:23:27 - [warn] Projects disabled : editorTheme.projects.enabled=false
26 Jan 11:23:27 - [info] Flows file     : \Users\E Bing Sen\.node-red\flows.json
26 Jan 11:23:27 - [info] Server now running at http://127.0.0.1:1880/
26 Jan 11:23:27 - [info] Starting flows
26 Jan 11:23:27 - [info] Started flows
```

Once the server has booted up successfully the command prompt will let us know that the server is now running, and what IP address it is on.

You can find it at the line - [info] Server now running at http://127.0.0.1:1880/.

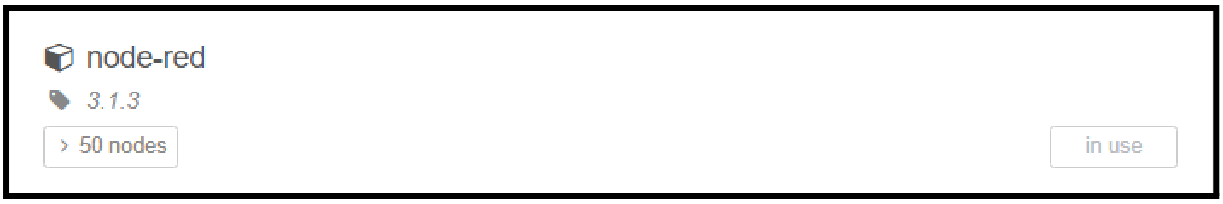
Now you can navigate to your web browser and input the IP address of the Node-RED web server or click on the link above to access the web server as most users will have the same IP address as the IP address of the machine running Node-RED. You have now successfully installed, booted up, and accessed your Node-RED web server.

You can now begin to create your own flows and install additional packages required for your flows.

Additional Packages downloads

The following are instructions on how to install additional packages in Node-RED.

By default, you should have a node palette downloaded called node-red. This module contains 50 nodes that are the basic building blocks for creating flows. All nodes include documentation you can see in the Info sidebar tab when you select a node.


node-red

3.1.3

The following additional packages are needed when following this integration guide.

node-red-dashboard

The first package you will need to install is the node-red-dashboard, this module adds nodes which allow us to easily create a live data dashboard. We will be using several nodes in this module for this guide.

To install this module, you have to:

1. Navigate to the hamburger menu on the top left of your screen.
2. Open the Manage palette tab. Alternatively (Alt+Shift+P).
3. Navigate to the Install tab.
4. Insert “node-red-dashboard” in the search bar.
5. Press the Install button.

node-red-contrib-ui-upload

The second package you will need to install is the node-red-contrib-ui-upload, this module adds nodes which allow us to upload a file content by WebSocket (Socket.io) streaming. We will be using the upload node in this guide.

To install this module, you have to:

1. Navigate to the hamburger menu on the top left of your screen.
2. Open the Manage palette tab. Alternatively (Alt+Shift+P).
3. Navigate to the Install tab.
4. Insert “node-red-contrib-ui-upload” in the search bar.
5. Press the Install button.

node-red-node-ui-table

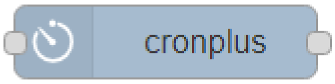
The third package you will need to install is the node-red-node-ui-table, this module adds nodes which allow us to display data as a table. We will be using the table node in this guide.

To install this module, you have to:

1. Navigate to the hamburger menu on the top left of your screen.
2. Open the Manage palette tab. Alternatively (Alt+Shift+P).
3. Navigate to the Install tab.
4. Insert "node-red-node-ui-table" in the search bar.
5. Press the Install button.

node-red-contrib-cron-plus

The fourth package you will need to install is the node-red-contrib-cron-plus, this module adds nodes which allow us to have full dynamic control with a flexible timer/scheduler. We will be using the cron-plus node in this guide.

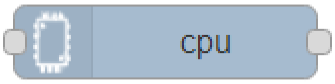


To install this module, you have to:

1. Navigate to the hamburger menu on the top left of your screen.
2. Open the Manage palette tab. Alternatively (Alt+Shift+P).
3. Navigate to the Install tab.
4. Insert "node-red-contrib-cron-plus" in the search bar.
5. Press the Install button.

node-red-contrib-cpu

The fifth package you will need to install is the node-red-contrib-cpu, this module adds nodes which allow us to monitor your cpu usage. We will be using the cpu node in this guide.

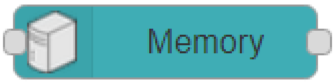


To install this module, you have to:

1. Navigate to the hamburger menu on the top left of your screen.
2. Open the Manage palette tab. Alternatively (Alt+Shift+P).
3. Navigate to the Install tab.
4. Insert "node-red-contrib-cpu" in the search bar.
5. Press the Install button.

node-red-contrib-os

The last package you will need to install is the node-red-contrib-os, this module adds nodes which allow us to monitor your cpu system information. But we will be only focusing on using the Memory node in this guide.

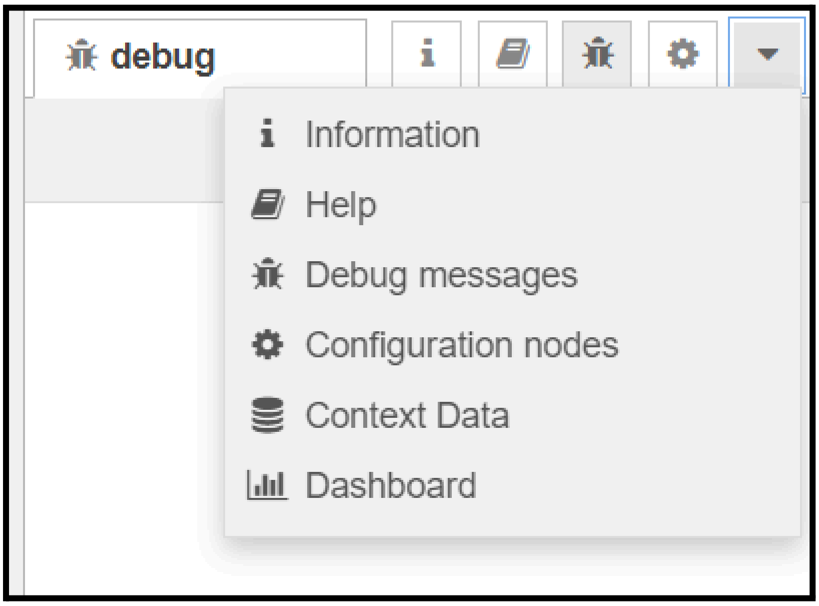


To install this module, you have to:

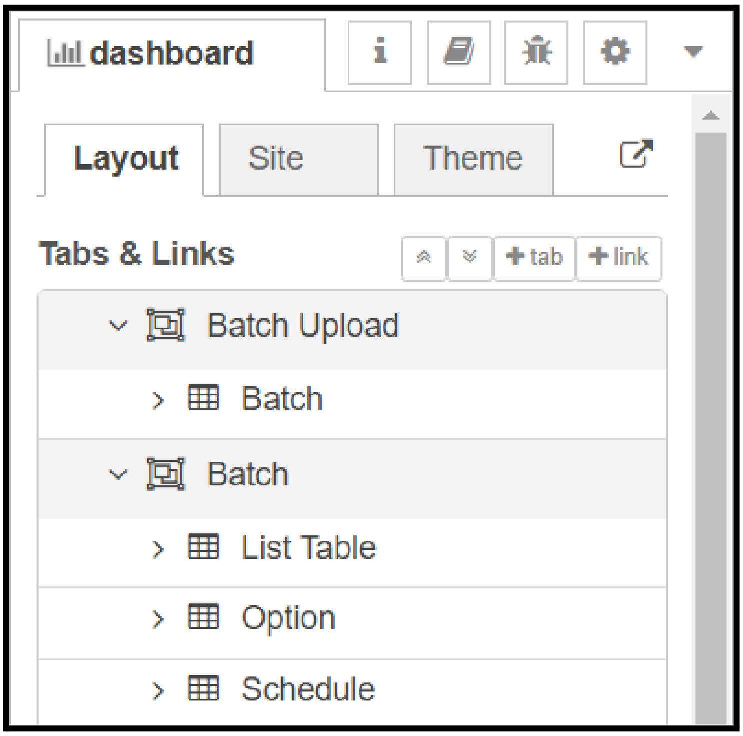
1. Navigate to the hamburger menu on the top left of your screen.
2. Open the Manage palette tab. Alternatively (Alt+Shift+P).
3. Navigate to the Install tab.
4. Insert “node-red-contrib-os” in the search bar.
5. Press the Install button.

Setting up the Dashboard

The following are instructions on setting up the dashboard and creating tabs for the Batch Processing Scheduler which includes Uploading the batch files and Scheduling the uploaded batch files.



You can access the dashboard by clicking on the dropdown button at the sidebar and navigate to the dashboard tab. You will need to create several tabs and groups for the Batch Processing Scheduler as it allows you to allocate any UI Nodes to be in their respective tabs. Example as shown below:

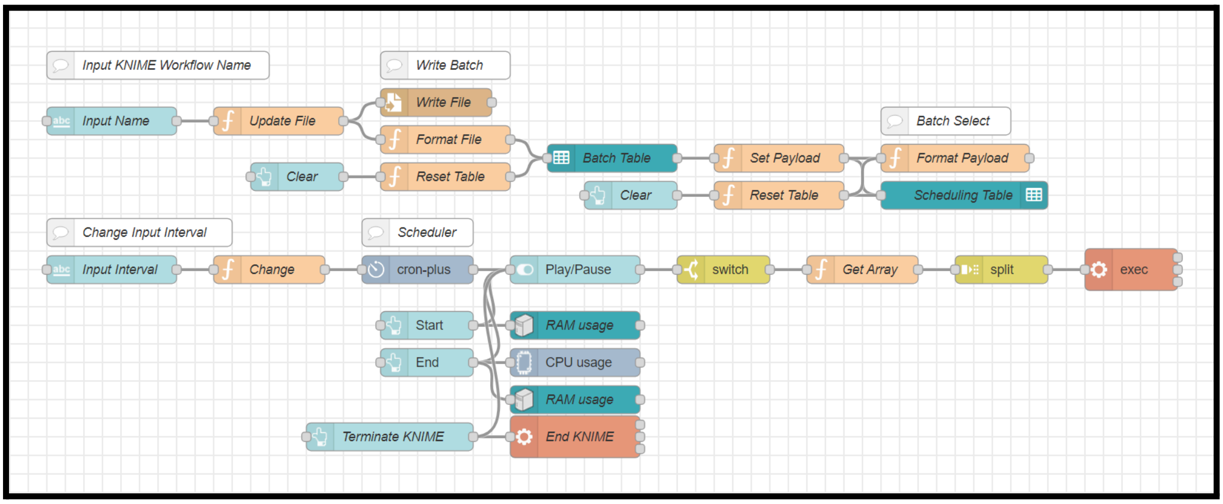


Downloading the Workflow

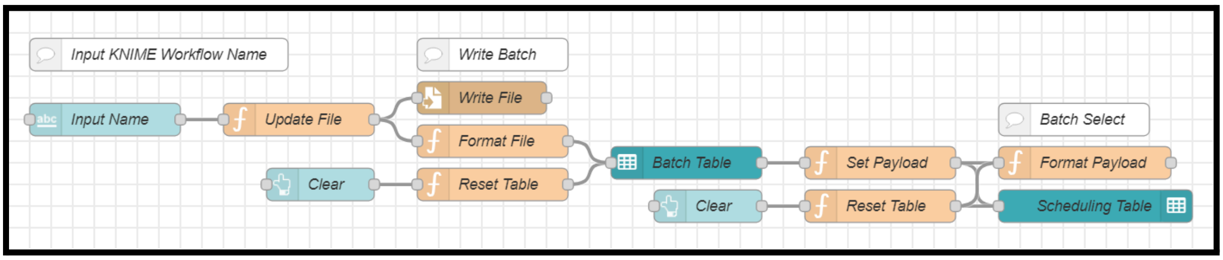
The following are instructions on how to download and import the workflow for the Batch Scheduling Process into Node-RED.

1. Go to the GitHub page for batch-processing-scheduler.
2. Download batchProcessingScheduler.json .
3. Open Node-RED and add flow.
4. Navigate to the hamburger menu.
5. Select Import (ctrl+i).
6. Then you can either paste flow json or upload a json file.

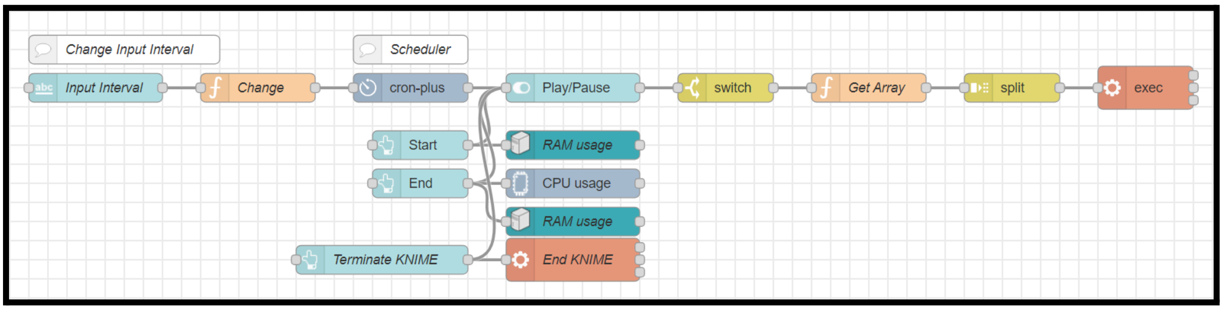
After importing the json, you should be able to see the flow as below.



Flow Explanation



In the initial part of the flow, a text input node is used to receive a string from users. This string is used to update and write a batch file into the designated directory. Once the batch file is written, its name is displayed in the Batch Table. Users can click on the file names listed in the Batch Table, and the selected filename will then be displayed in a secondary table, known as the Scheduling Table. For memory management, two 'Clear' buttons are provided to empty their respective tables when needed.



In the other part of the flow, a text input node is used to accept an integer from users, which updates the scheduling time in minutes. You can then click on the 'Start' or 'End' button to schedule the batch file appearing in the Scheduling Table. If you wish to monitor RAM and CPU usage, you can utilise a function node and incorporate it into a separate table for display. Additionally, a 'Terminate' button is available to close all KNIME instances for memory cleanup.

Important Nodes to Modify

Update File (Function Node)

```
let tableData = msg.payload;

let filenames = [];

for (let i = 0; i < tableData.length; i++) {
    let row = tableData[i];
    filenames.push(row.Filename);
}

msg.payload = filenames.map(file => `D:/BatchFiles/${file}`);

flow.set("Array", msg.payload);

return msg;
```

This Update File Function node uses a javascript code to change the filename to your desired file path so that the nodes will be able to access the batch file stored in your desired file path. **Please note** that `'D:/BatchFiles/'` will be the directory of where you will save these files. You should replace this with your preferred path. After making this change, the system will be able to locate and access the batch files in your chosen directory.

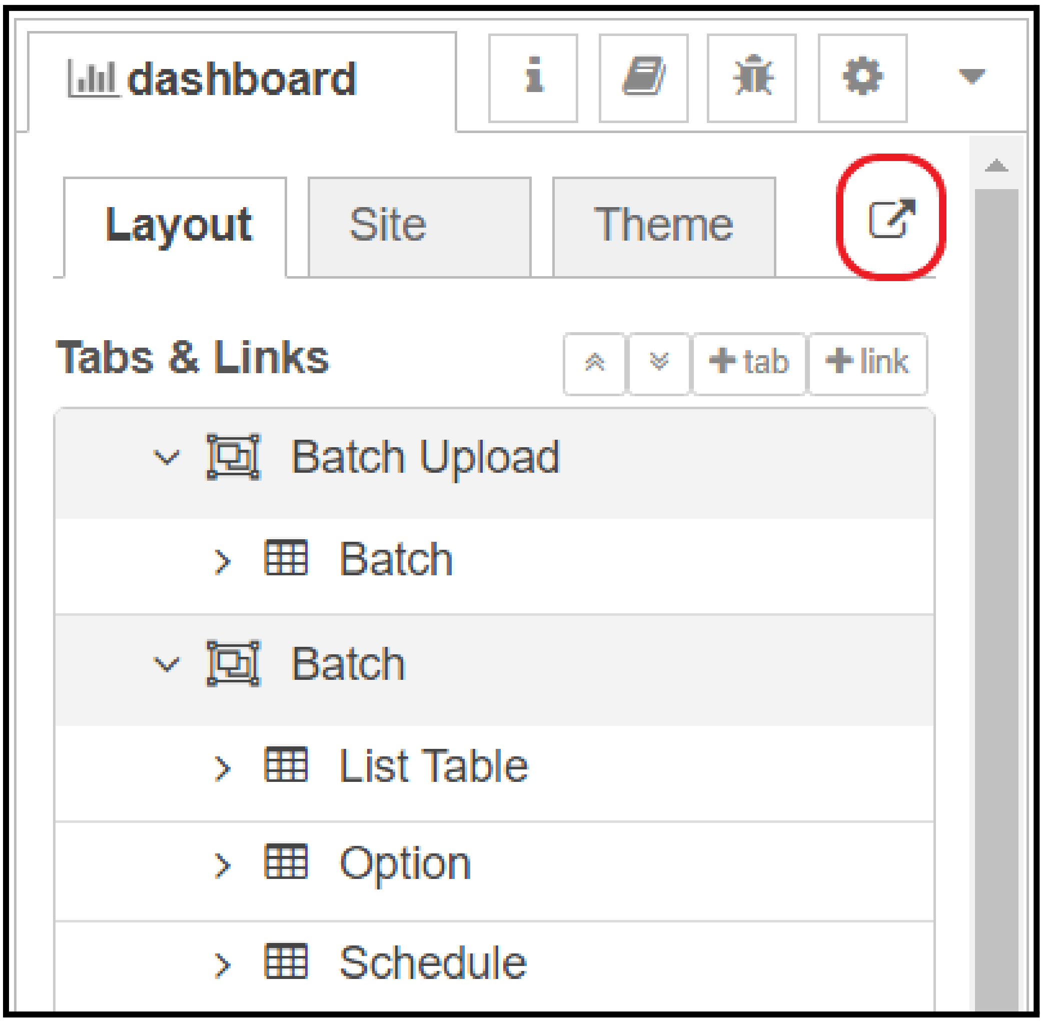
Format Payload (Function Node)

```
let tableData = msg.payload;

let filenames = [];

for (let i = 0; i < tableData.length; i++) {
    let row = tableData[i];
    filenames.push(row.Filename);
}

msg.payload = filenames.map(file => `D:/BatchFiles/${file}`);

flow.set("Array", msg.payload);

return msg;
```

This Format Payload Function node formats the name of the batch file from the table to obtain the path of the batch files stored previously as you uploaded them. **Please note** that `'D:/BatchFiles/'` will be the directory of where you will save these files. You should replace this with your preferred path. After making this change, the system will be able to access the batch files in your chosen directory.

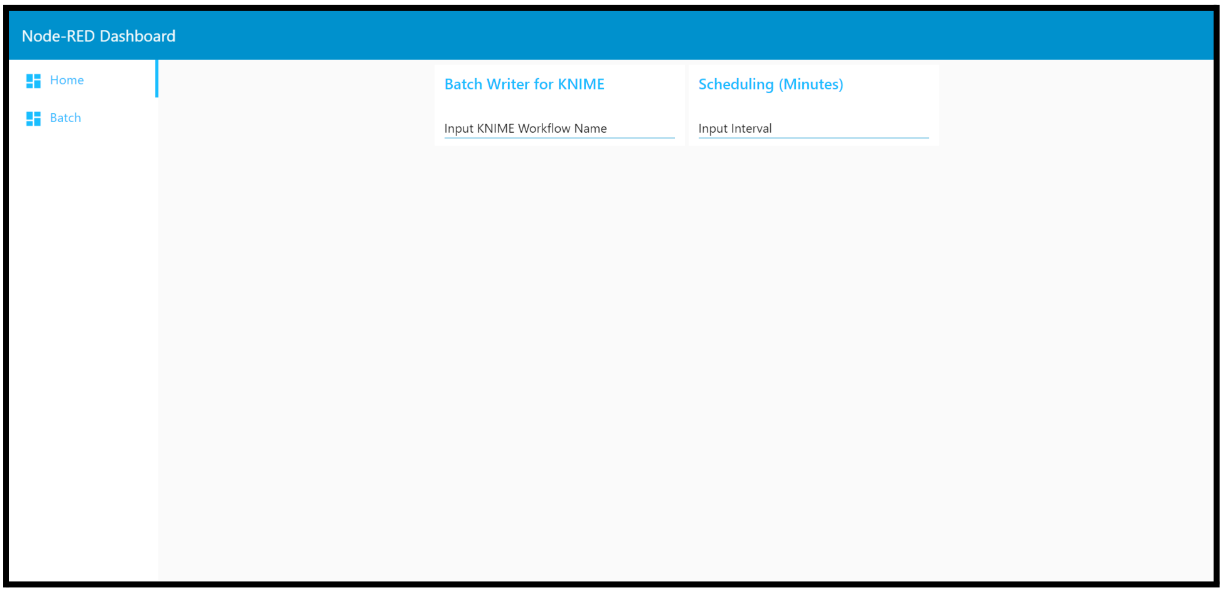
Deploy

Once the flow is successfully built, click on the ‘Deploy’ button to deploy the flow. Afterwards, navigate to the Node-RED Dashboard by clicking the indicated button.

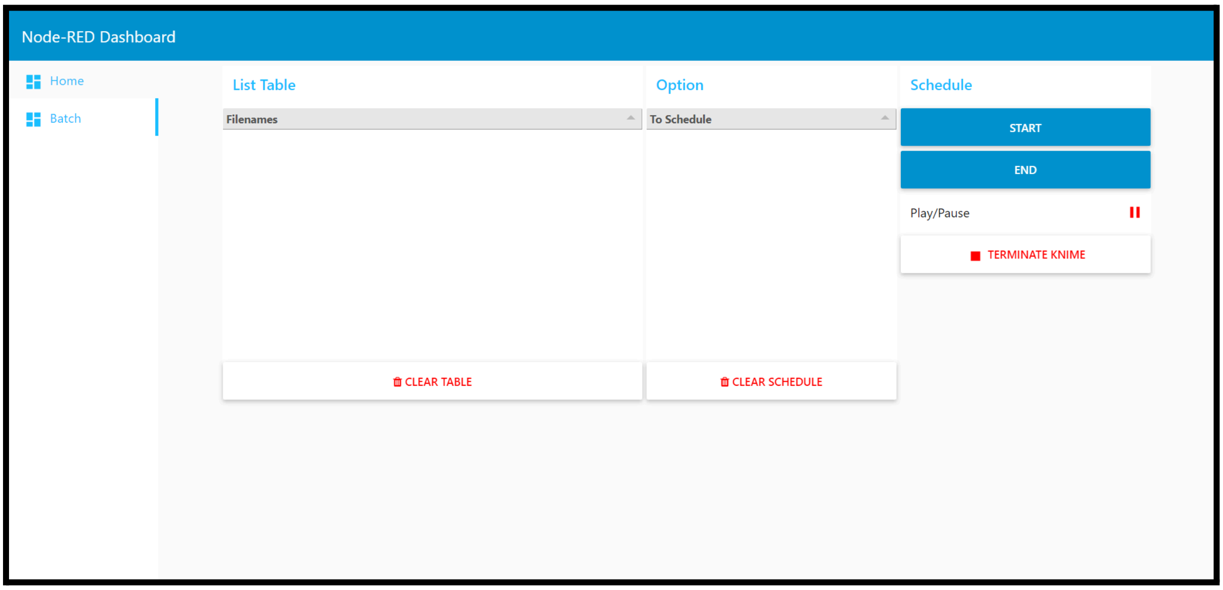


This will bring you to a new tab for the Node-RED Dashboard that looks like this:

First Tab



Second Tab



KNIME Workflows

The following are the instructions for configuring the KNIME workflows. You will be using them in this Batch Processing Scheduler. There is an example of a KNIME workflow in the GitHub page you could download, or you can create one using KNIME Analytics Platform.

Here is the link to download it: https://www.knime.com/downloads.

User Guide

The following guides the user on how to use the software to schedule batch processes in Node-RED Dashboard. Please refer to **Downloading the Workflow** for the setup and installation of the software.

Using the scheduler

1. Input Workflow Name and Interval:

Start by entering your KNIME workflow name and the interval for scheduling. This information will be automatically saved into the nodes.

2. Navigate to the Second Tab:

Move to the second tab in the interface. Here, you will find two tables:

1. The first table displays all the batch files added.
2. The second table shows batch files that have been selected for scheduling.

3. Select Batch Files for Scheduling:

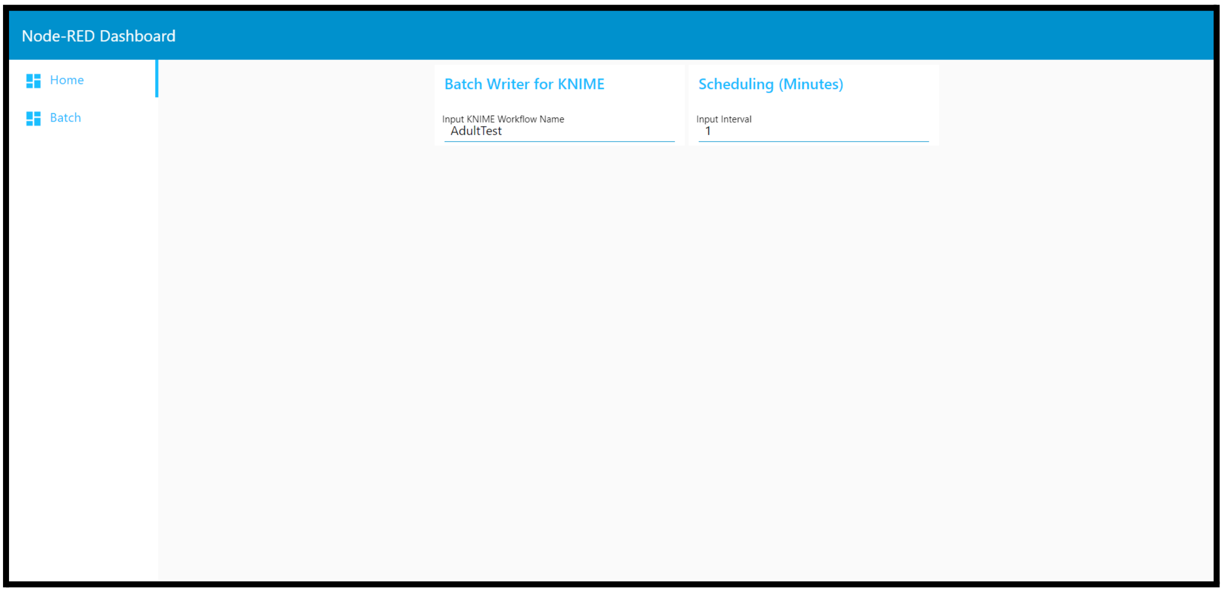
To select batch files for scheduling, follow these steps:

1. Click on the batch files you want to schedule in the first table.
2. The selected batch files will then appear in the second table.

4. Initiate Scheduling Process:

Once you have chosen the batch files for scheduling, a hidden group will appear. This group contains Start and Stop buttons for initiating and ending the scheduling process.

Example usage:

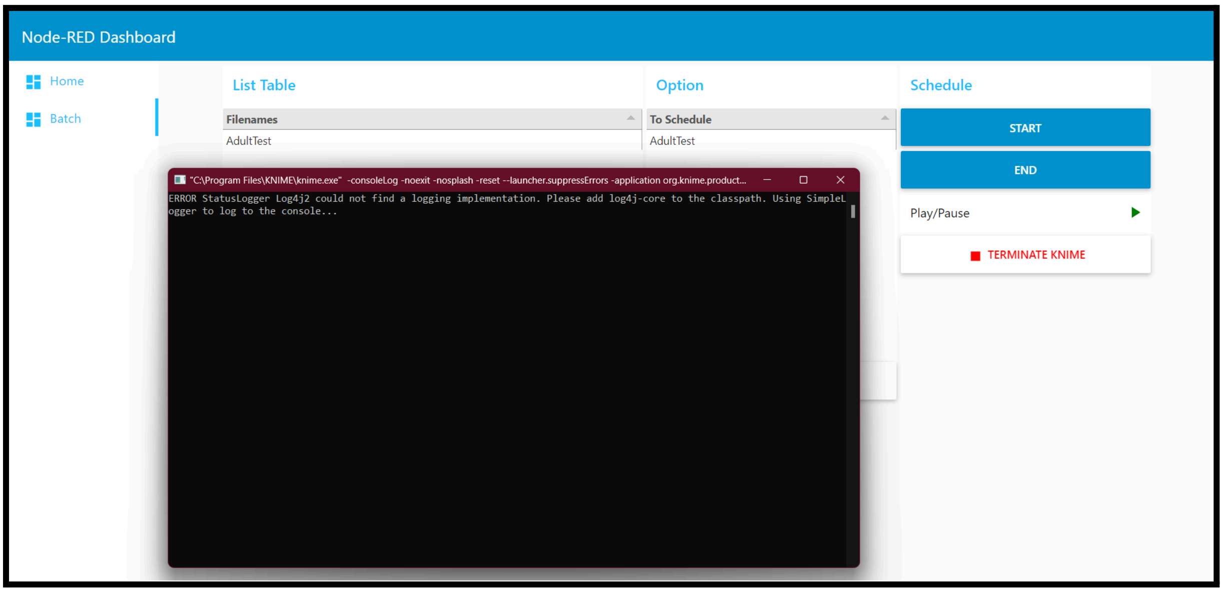


Batch Writer for KNIME

Input KNIME Workflow Name: AdultTest

Scheduling (Minutes)

Input Interval: 1



List Table

Filenames
AdultTest

Option

To Schedule
AdultTest

Schedule

```
"C:\Program Files\KNIME\knime.exe"  -consoleLog -noexit -nosplash -reset --launcher.suppressErrors -application org.knime.product...
ERROR StatusLogger Log4j2 could not find a logging implementation. Please add log4j-core to the classpath. Using SimpleL
ogger to log to the console...
```

This line marks the end of this document