

1. Next-Generation Workflow (NGW)	2
1.1 Basic Workflow Tasks	3
1.1.1 Add a file to the workflow canvas	4
1.1.2 Adding nodes to a workflow	5
1.1.3 Creating a workflow	8
1.2 Creating and running a simple workflow to run Sierra (CEE)	10
1.2.1 Adding parameters and responses to a Sierra workflow	14
1.3 Workflow "Hello, World" tutorial	16

Next-Generation Workflow (NGW)

Introductory Tutorials

- [Workflow "Hello, World" tutorial](#)
- [Creating and running a simple workflow to run Sierra \(CEE\)](#)
 - [Adding parameters and responses to a Sierra workflow](#)

Basic Workflow Tasks

- [Creating a workflow](#)
- [Adding nodes to a workflow](#)
- [Add a file to the workflow canvas](#)

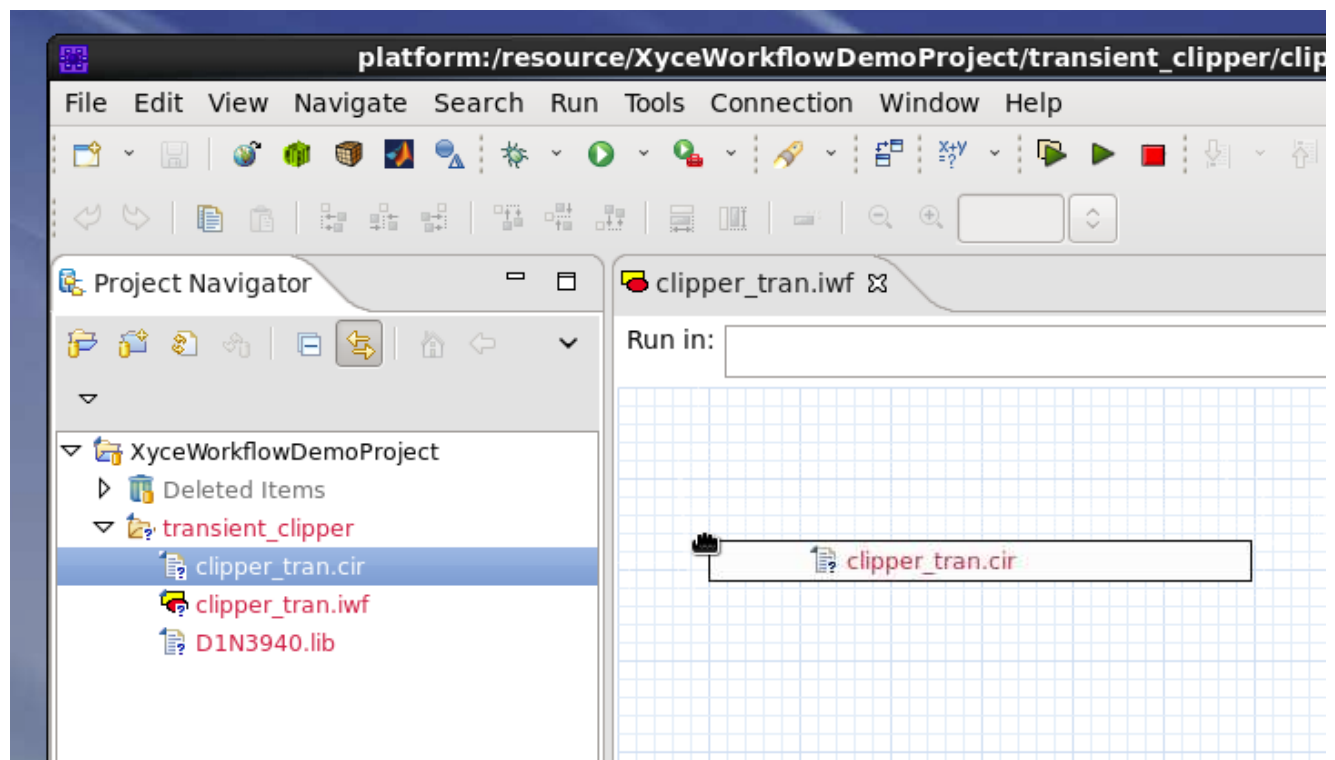
Basic Workflow Tasks

- [Add a file to the workflow canvas](#)
- [Adding nodes to a workflow](#)
- [Creating a workflow](#)

Add a file to the workflow canvas

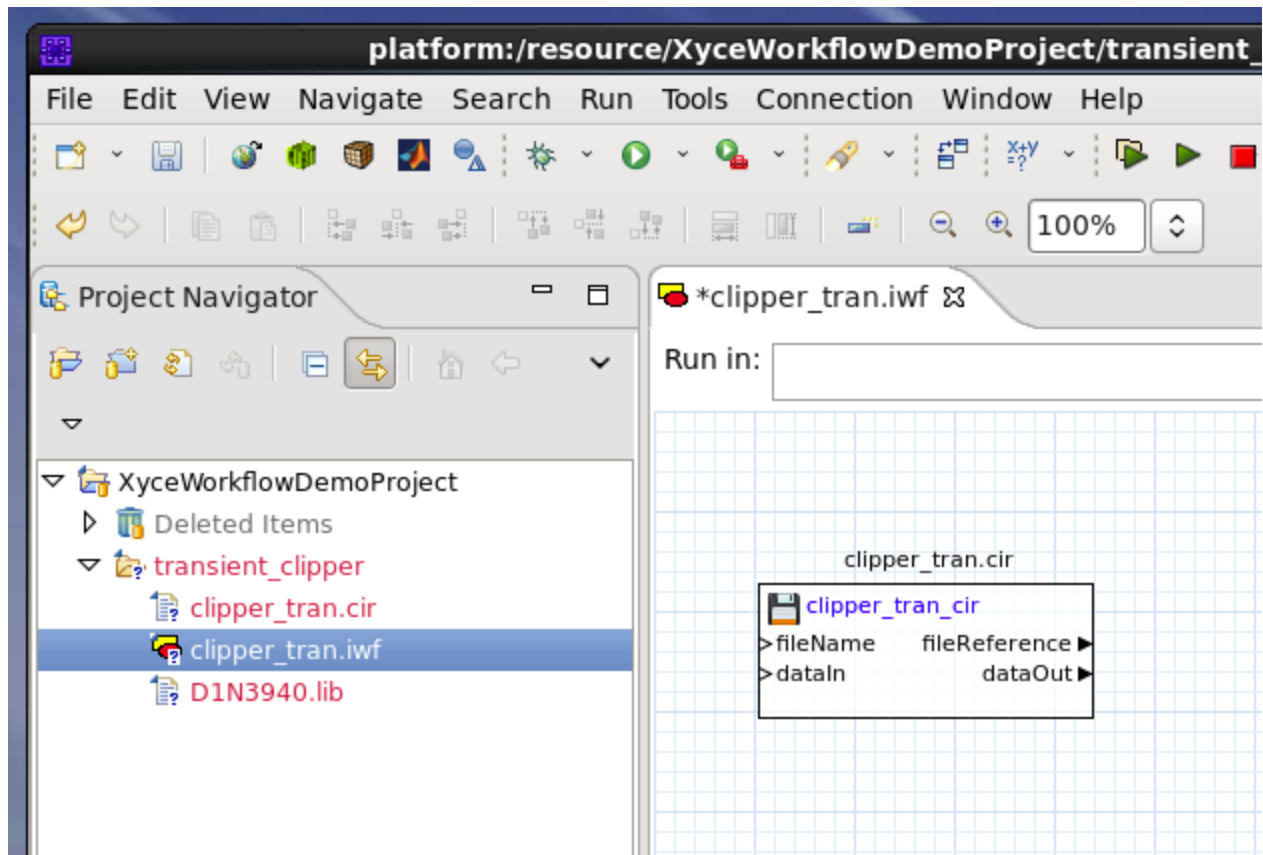
What to do

Click on a file in the Project Navigator and drag it to the workflow canvas.



What happens

A file node corresponding to the file you dropped is added to the workflow.

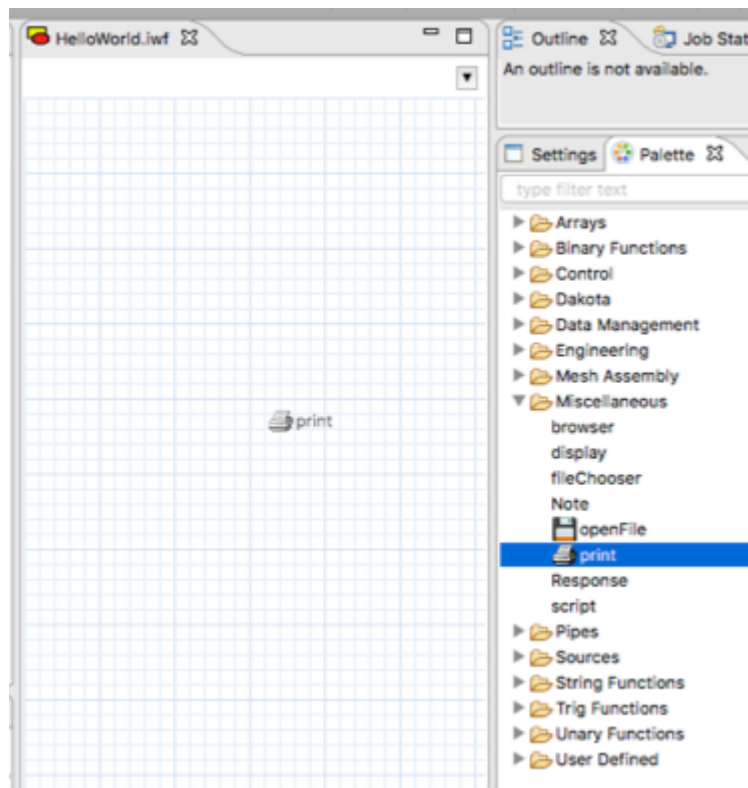


Adding nodes to a workflow

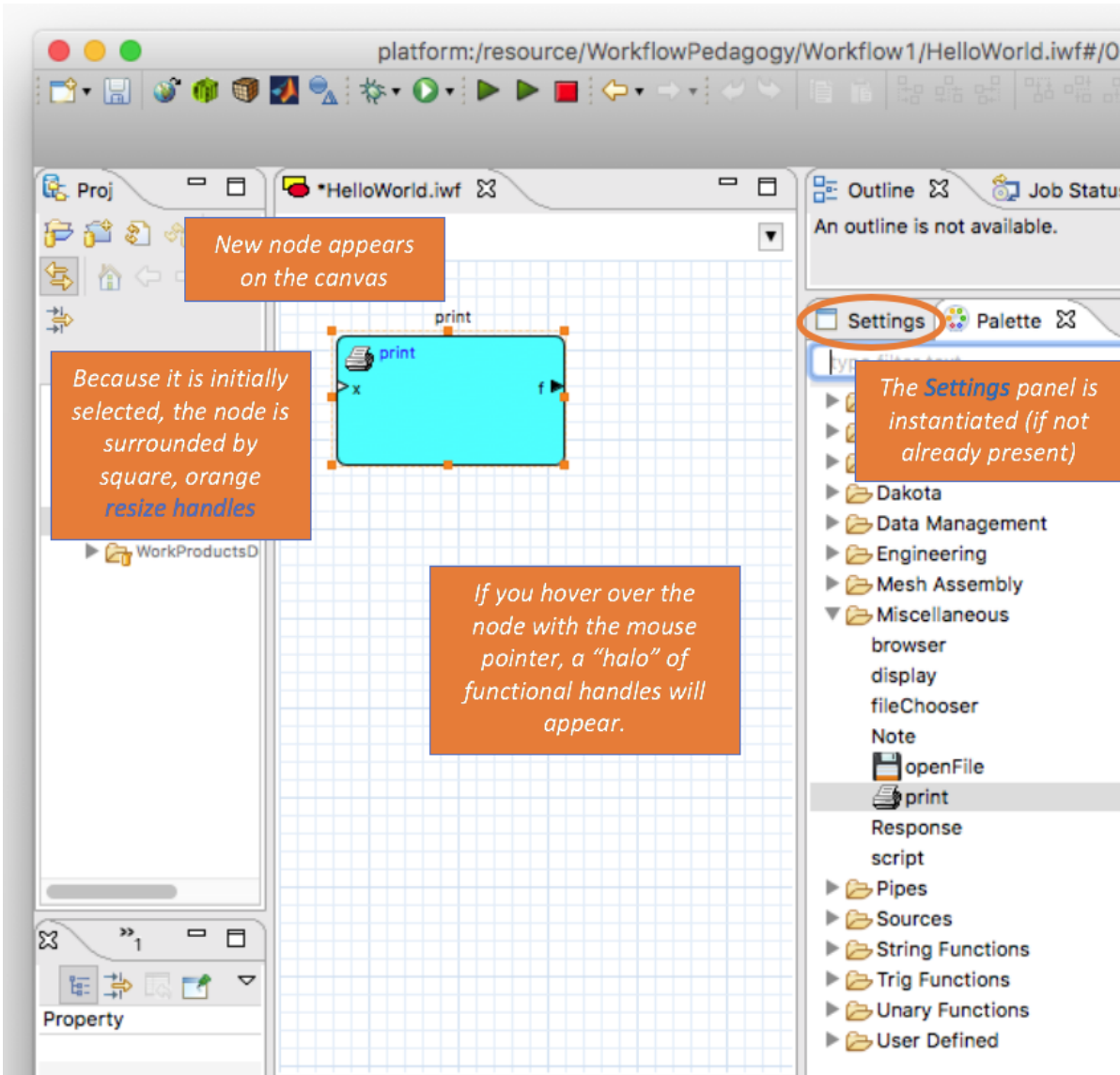
rkflow

Steps

1. Expose available node types in the Palette view by either (a) expanding one or more of the listed categories, or (b) typing into the search box.
2. Click on the desired node type and drag it onto the workflow canvas.



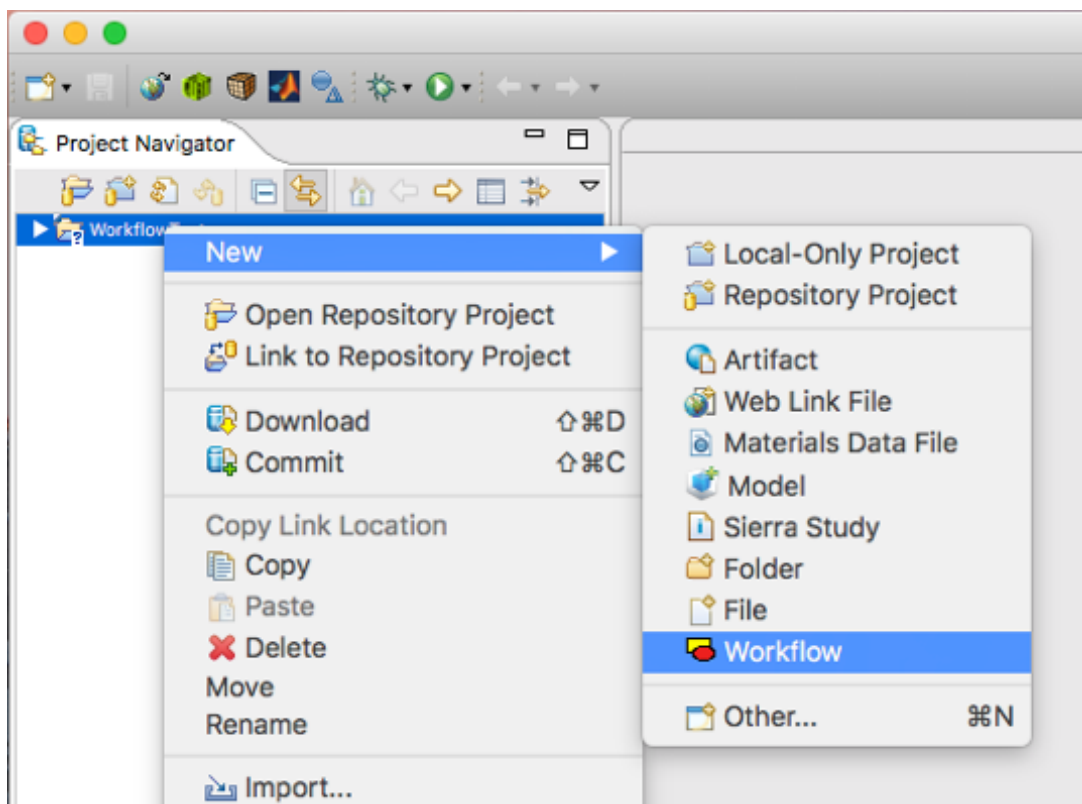
What should happen



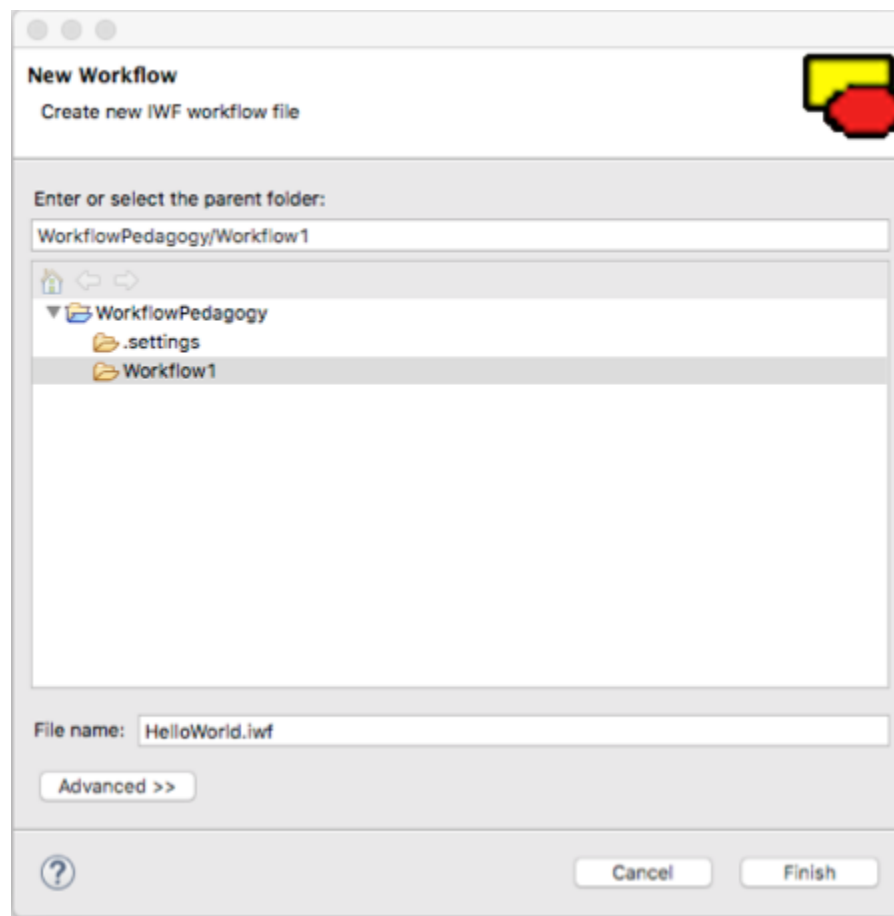
Creating a workflow

What to do:

1. Choose File New Workflow

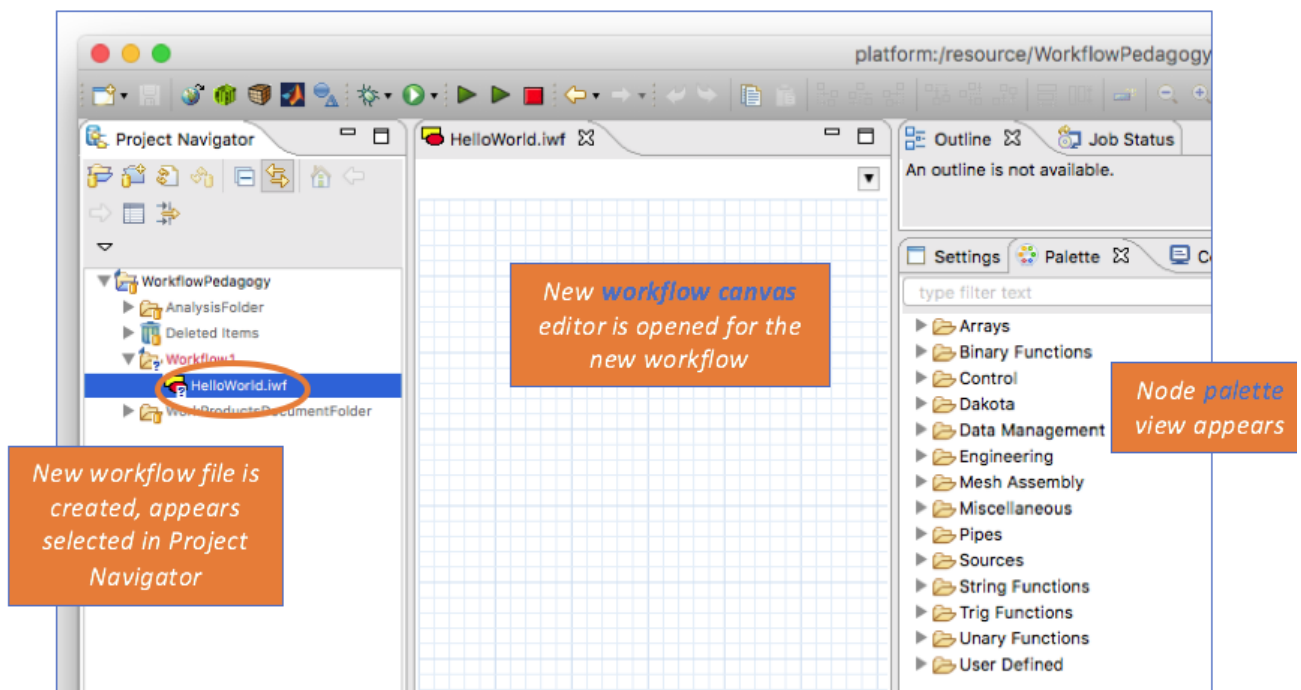


2. Designate the parent folder for the workflow (if one is not already selected), and enter a name for the new workflow file (a `.iwf` suffix will be appended).



3. Click on "Finish"

What happens:



Creating and running a simple workflow to run Sierra (CEE)

This brief tutorial takes you through the steps necessary to run a simple Sierra workflow on a CEE LAN machine (i.e. running Sierra directly, without queue submission).

Prerequisites

To complete the steps below, you will first need to:

1. Launch an alpha build of SAW

See instructions [here](#) if you need to launch an alpha build of SAW for the first time on a CEE LAN machine

2. Have at least one project open in SAW's **Project Navigator**

See instructions [here](#) to open a SAW repository project

3. Add the following two files to a project in SAW's **Project Navigator**:



See [here](#) for an example

Steps

1. [Create a new workflow](#) in the project folder containing the **CB.jou** and **CB.i** files.

2. Drag-and-drop **CB.jou** and **CB.i** from SAW's **Project Navigator** to the workflow canvas.

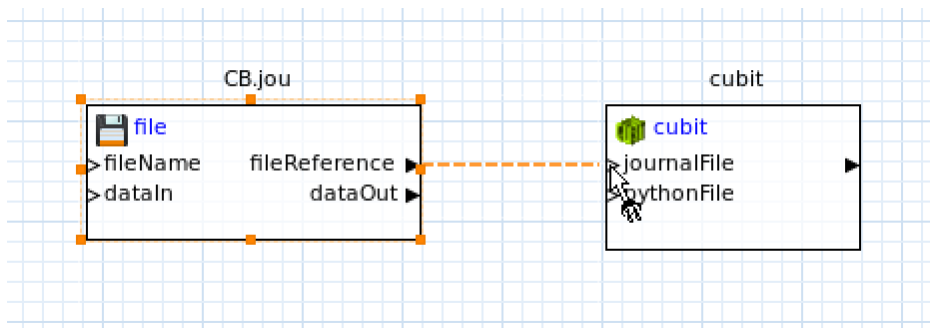
[Add a file to the workflow canvas](#)

3. Add a **cubit** node to the workflow canvas.

[Adding nodes to a workflow](#)

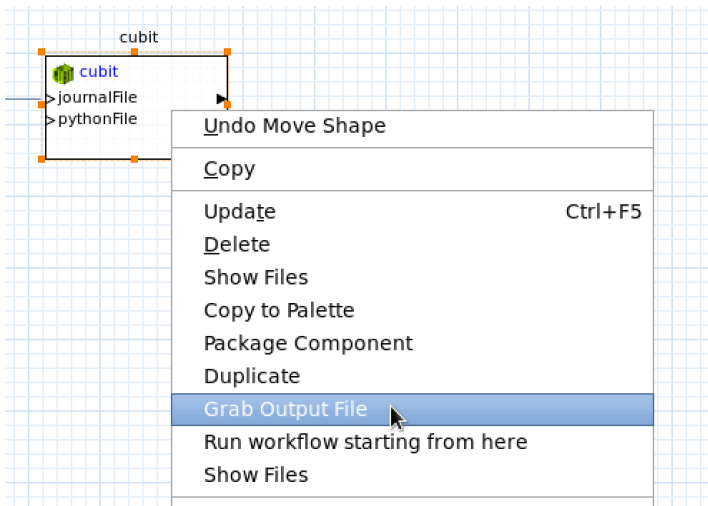
4. Connect Cubit journal (**CB.jou**) file node to the **cubit** node

a. Click on the **fileReference** output port of the **CB.jou** file node, drag over to the **journalFile** input port on the **cubit** node, and release to make the connection.

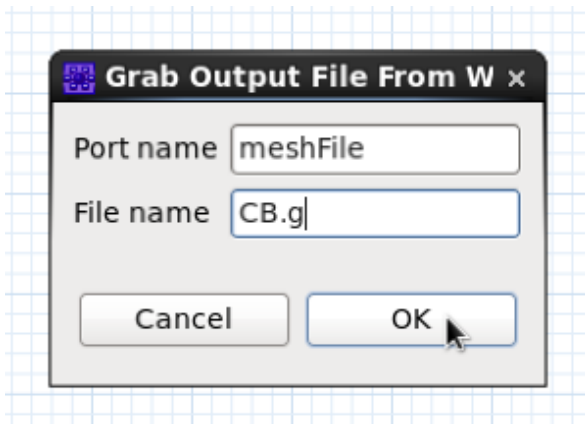


5. Add an output port for the mesh file to the **cubit** node.

a. Right-click on the **cubit** node and choose **Grab Output File**.



b. In the resulting pop-up, enter `meshFile` for the **Port name**, `CB.g` for the **File name**, and click **OK**.

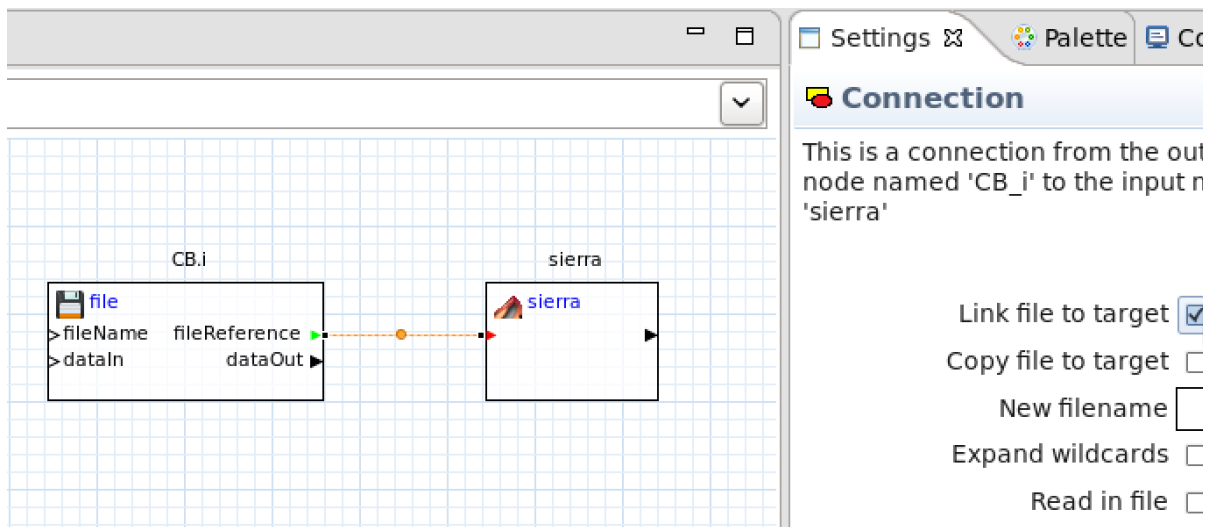


5. Add a **sierra** node to the workflow canvas.

6. Connect the Sierra input deck (**CB.i**) file node to the **sierra** node.

a. Click on the **fileReference** output port of the **CB.i** file node, drag to the input port on the **sierra** node, and release to create the connection.

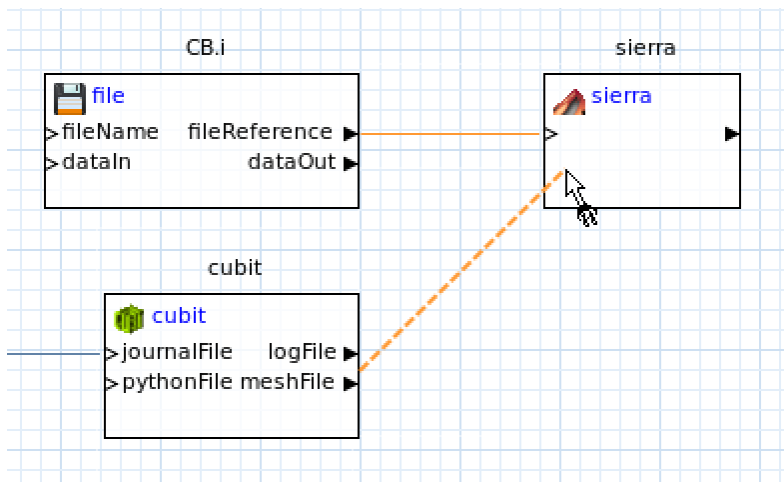
b. With the connection between the **CB.i** file node and the **sierra** node selected, go to the **Settings** panel and check **Link file to target**.



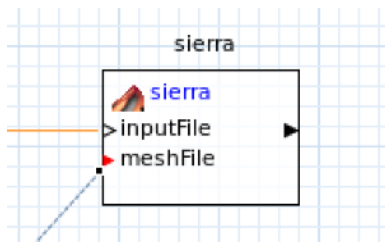
Selecting **Link file to target** here causes the Sierra input deck to be linked into the **sierra** node's *private work directory* where Sierra will be executed.

7. Connect the **meshFile** output port of the **cubit** node to (what will be) a new input port on the **sierra** node.

a. Click on the **meshFile** output port of the **cubit** node and drag over to the **sierra** node, but this time release the connection when hovering over empty space in the body of the **sierra** node.



This will result in a new input port being created for the connection.

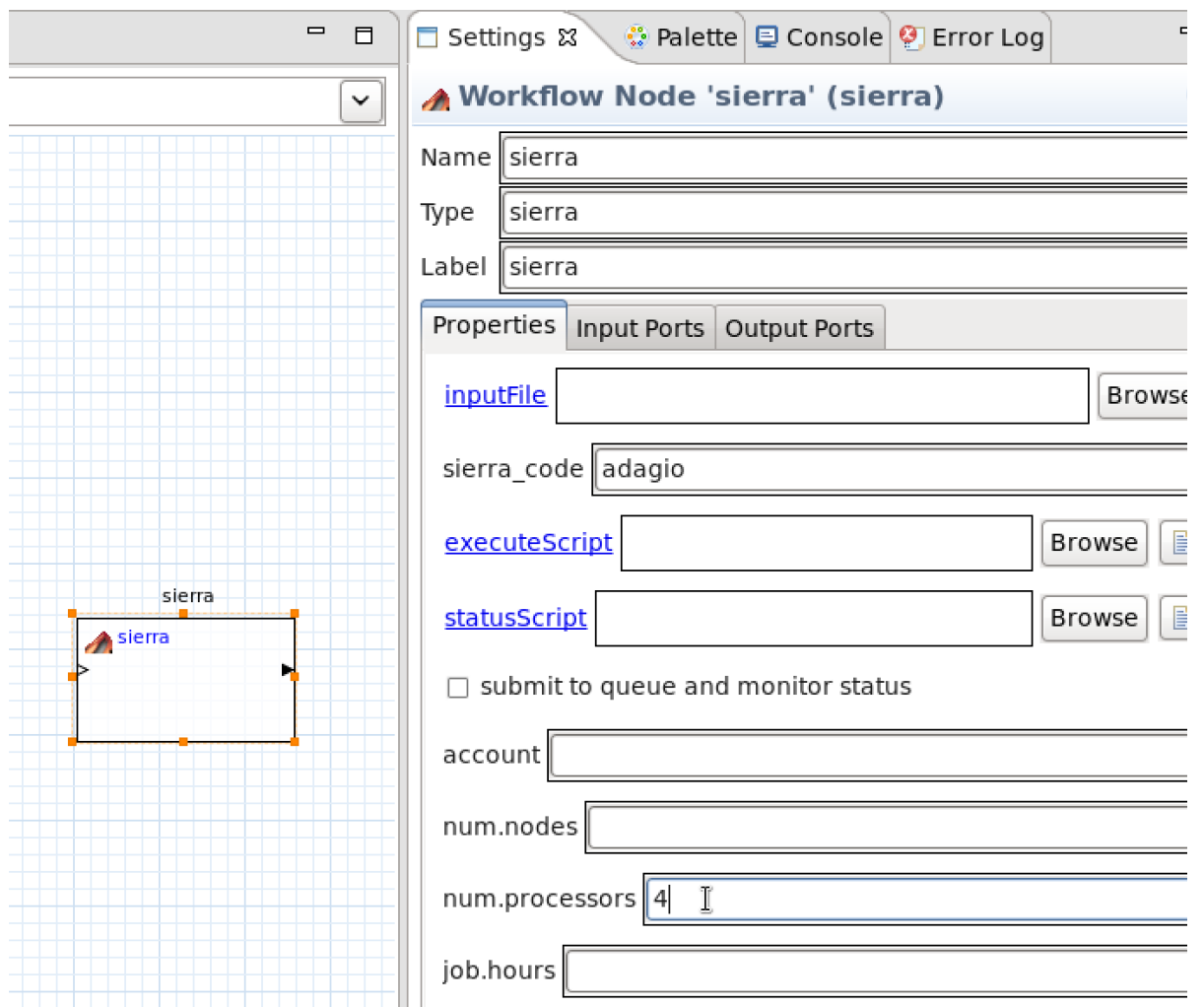


b. With the connection between the **cubit** node and the **sierra** node selected, go to the **Settings** panel and check **Link file to target**.

8. Configure the **sierra** node.

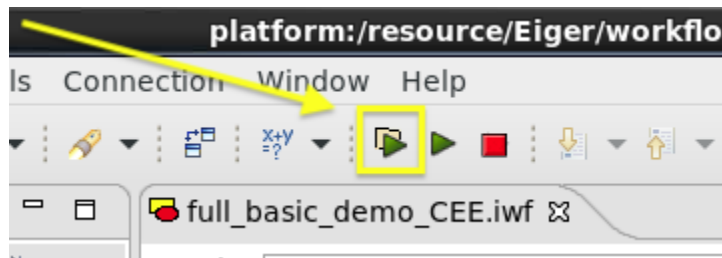
a. With the **sierra** node selected, click on the tab for the **Settings** panel to bring it to the foreground.

b. In the **Properties** tab, enter `adagio` in the **sierra_code** field, un-check **submit to queue and monitor status**, and enter 4 in the **num. processors** field.



9. Execute the workflow.

Click on the left-most of the two "play" buttons in the main toolbar, and then click on "Finish" in the "Choose Directory" pop-up.



Adding parameters and responses to a Sierra workflow

This tutorial provides an example of

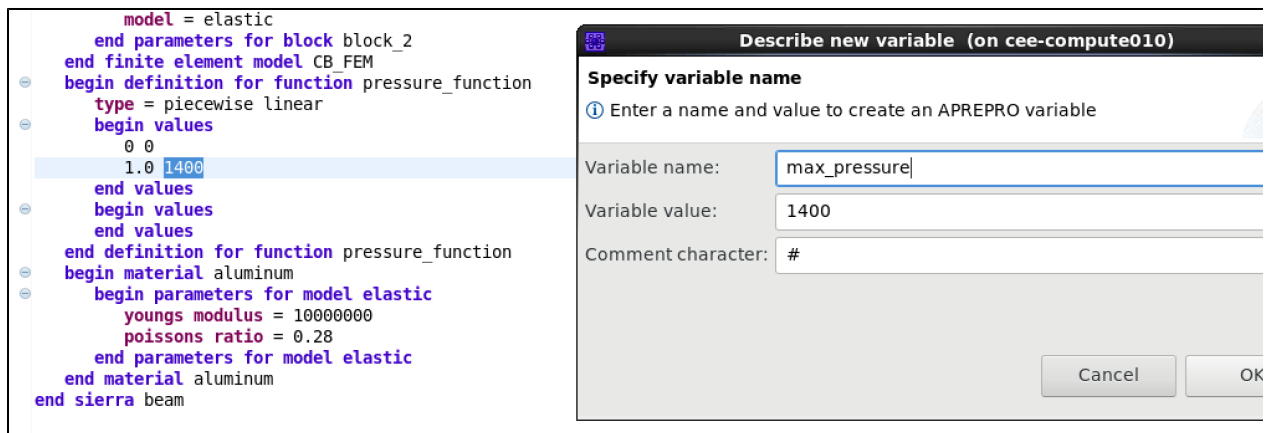
- Exposing a quantity affecting the simulation workflow as a parameter that may be varied by external means.
- Exposing a quantity of interest computed by the simulation as an externally accessible response.

Prerequisites

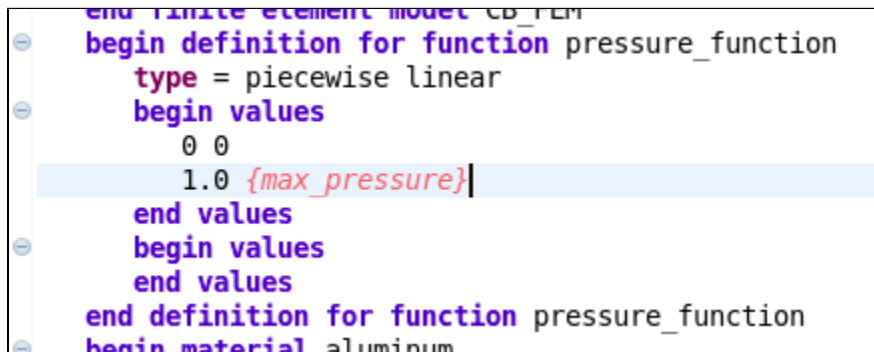
You should complete [Creating and running a simple Sierra workflow](#) before attempting the steps below.

Specifying a parameter in the Sierra input deck

1. Starting from the project where [Creating and running a simple Sierra workflow](#) was completed, double-click on **CB.i**, the Sierra input file, in SAW's **Project Navigator** to open it in an editor view.
2. In the editor view for **CB.i**, scroll down to near the bottom of the file and find the section that starts with `begin definition for function pressure_function`
3. Select the final pressure value (**1400**), right-click, and choose **Create a parameter**.
- x4. In the pop-up dialog box that appears, enter `max_pressure` as the name for the parameter ("variable") being defined and then click **OK**.



The numeric value will be replaced by the name of the parameter, `max_pressure`, surrounded by curly braces (the syntax used by pre-processor, `aprepro`).



Add the parameter to the workflow

Next, add a `max_pressure` parameter at the level of the workflow, and then configure the workflow to substitute the value specified for the parameter into the Sierra input deck at runtime.

1. Open the **CB.iwf** workflow for editing.
2. Add a **parameter** node to the workflow.

In **Settings**, set the name to `max_pressure` and the default value to `1400`.

3. Add an **aprepro** node to the workflow.

The **aprepro** node will substitute the value specified for the parameter into the input deck.

4. Connect in the **aprepro** node

- a. Select the connection between the **file** node for **CB.i** and the **inputFile** port on the **sierra** node and delete the connection (either by hitting the delete key or by right-clicking and choosing **Delete**).
- b. Connect the **fileReference** output port of the **file** node for **CB.i** to the **templateFile** input port of the **aprepro** node.
- c. Connect the output port of the **aprepro** node to the **inputFile** port of the **sierra** node.

Add a response to the workflow

The input deck used here directs Sierra to produce a heartbeat data file containing quantities of interest.

1. Add a port to the **sierra** node to send heartbeat data downstream.

- a. Right-click on the **sierra** node and choose **Grab Output File**.
- b. In the **Grab Output File** dialog that appears, enter **heartbeatFile** for **Port name** and **CB.aprepro.hrt** for **File name** and click **OK**.

2. **heartbeat** node

- a. Add a **heartbeat** node to the workflow
- b. With the **heartbeat** node selected, go to the **Properties** tab of the **Settings** view.
- c. Click to the right of **function** and choose **min value of column** from the drop-down.
- d. Enter **end_displ** in value field for the **columnX** property.

3. **response** node

- a. Add a **response** node to the workflow.
- b. In **Settings** (with **response** node selected), change the value of **Name** from **r** (the default value) to **displacement**.

4. Connect in the new nodes.

- a. Connect the **heartbeatFile** port of the **sierra** node to the input port of the **heartbeat** node.

After making the connection, go to **Settings** for the connection and select the **Read in file** action (under Other connection actions).

- b. Connect the output port of the **heartbeat** node to the response node now named **displacement**.

=====

Workflow "Hello, World" tutorial

These instructions lead you through building a minimal, 1-node workflow and then running it.

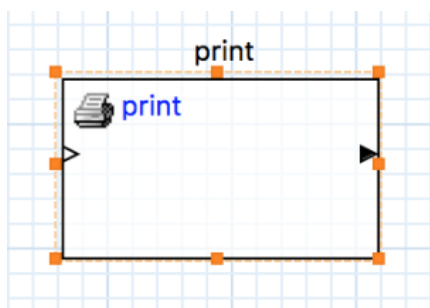
Steps

1. [Create a new workflow](#)

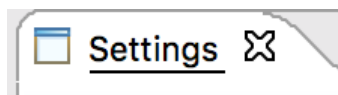
2. Add a **print** node to your workflow

3. Configure the **print** node

a. Click on the **print** node on the workflow canvas to select it.

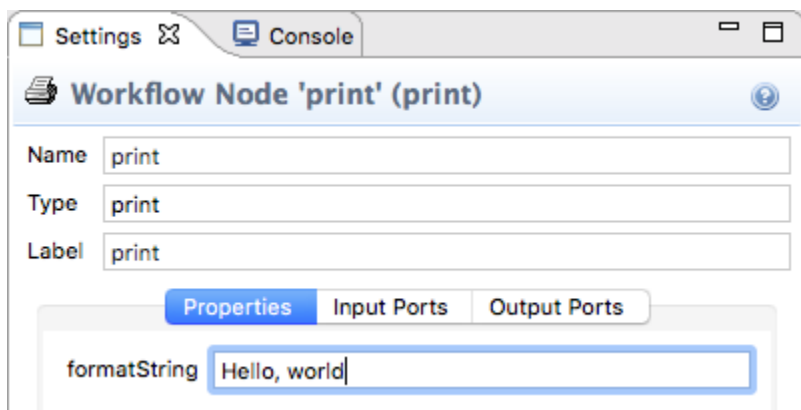


b. Find the tab for the **Settings** panel and click on it to bring it to the foreground.




(Note: After bringing the **Settings** panel to the foreground, you *might* have to click on the **print** node itself one more time to get the **print** node's settings to show in the **Settings** panel.)

c. Enter `Hello, world` in **formatString** field on the **Properties** tab of node **Settings** panel.

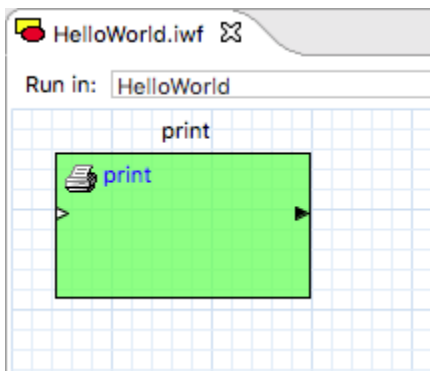


4. Execute the workflow from within SAW.

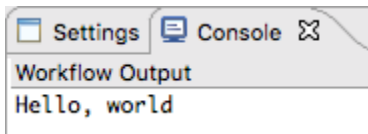
a. Click on the  button in the toolbar near the top of the SAW window to run the workflow.

b. Notice what happens.

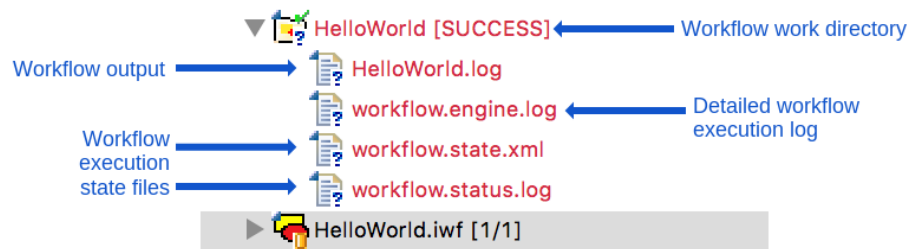
The **print** node turns green, indicating that it has been successfully executed.



Hello, world is output in the **Console** view.



In the **Project Navigator**, you can see that a work directory (named for the workflow, by default) has been created for this execution of the workflow, within which are a set of files that are generated in the course of workflow execution.



5. Execute the workflow from the command-line (on Linux, MacOS, or Windows):

- Change your working directory to the one where the .iwf file resides.
- Create a sub-directory that will be the work directory for this execution of the workflow, and `cd` into this subdirectory.
- Execute the workflow by entering:

`~/ .dart/runNgw.sh HelloWorld.iwf` (on Linux or MacOS)

or

`%userprofile%\ .dart\runNgw.bat HelloWorld.iwf` (on Windows)

d. Notice what happens

As in the case of embedded execution above, the files `workflow.engine.log`, `workflow.state.xml`, and `workflow.status.log` are all created, this time in the current working directory, the work directory for this execution.

In contrast with embedded execution, no `HelloWorld.log` file is created; `Hello, world` simply appears as output from running the command.