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 Workflow "Hello, World" tutorial 10

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

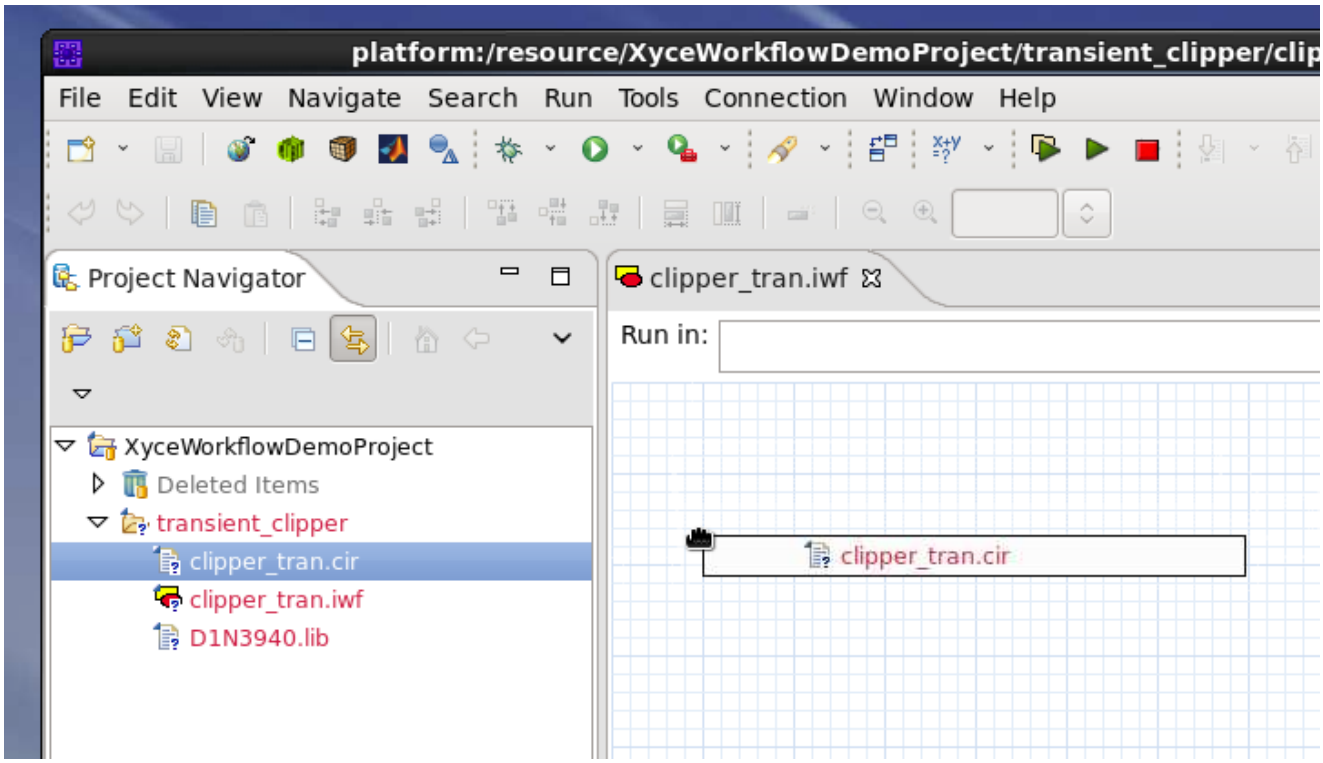
Basic Workflow Tasks

- Add a file to the workflow canvas
- Adding nodes to a workflow
- Creating a workflow

Add a file to the workflow canvas

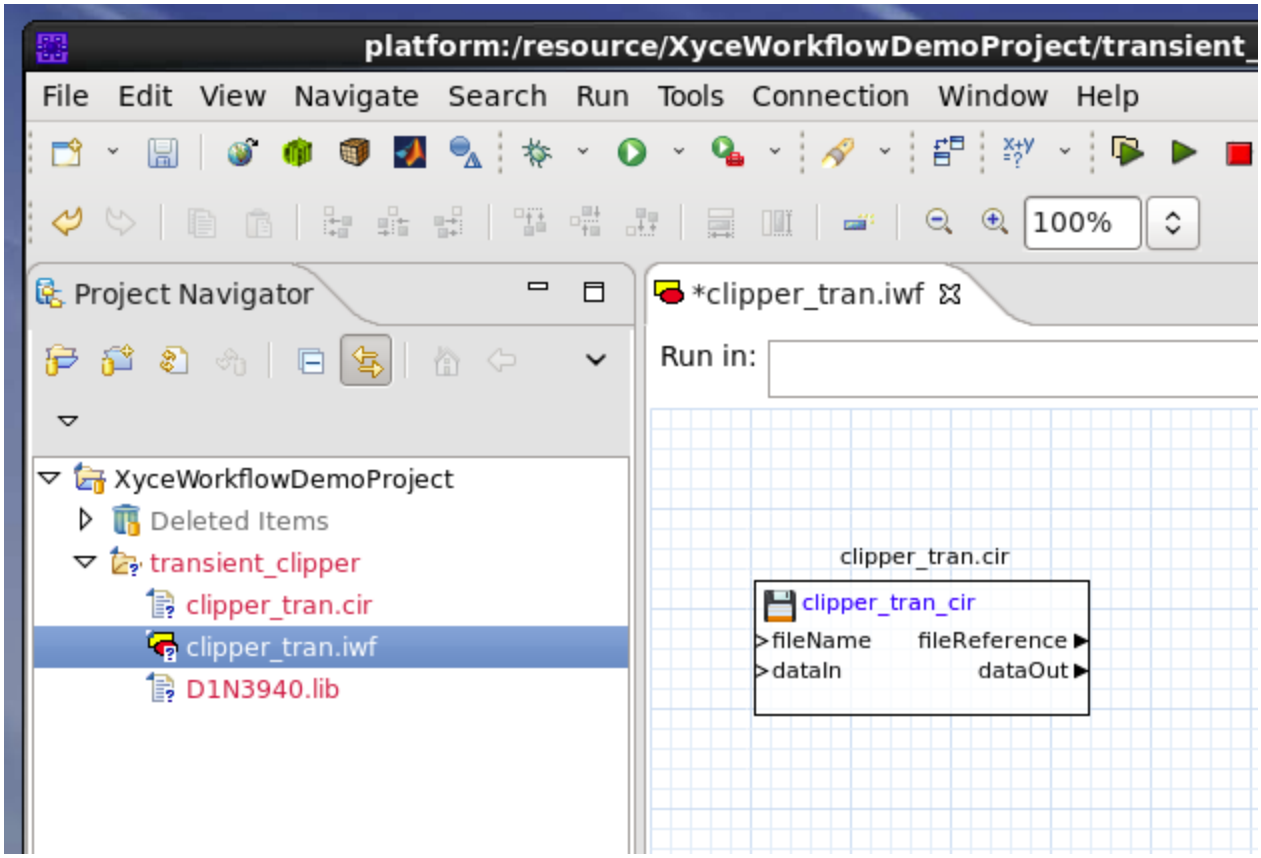
What to do

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



What happens

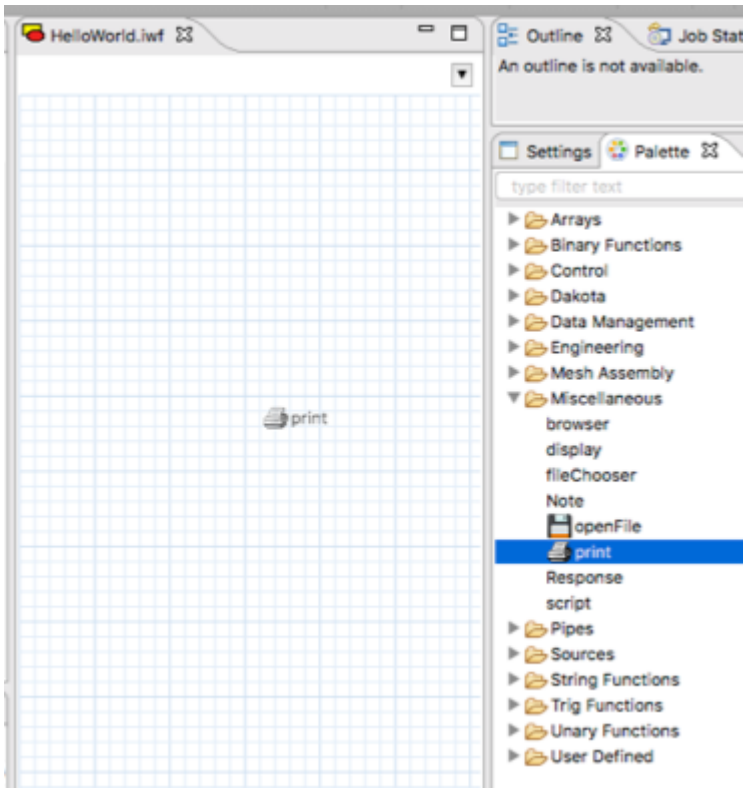
A file node corresponding to the file you dropped is added to the workflow.



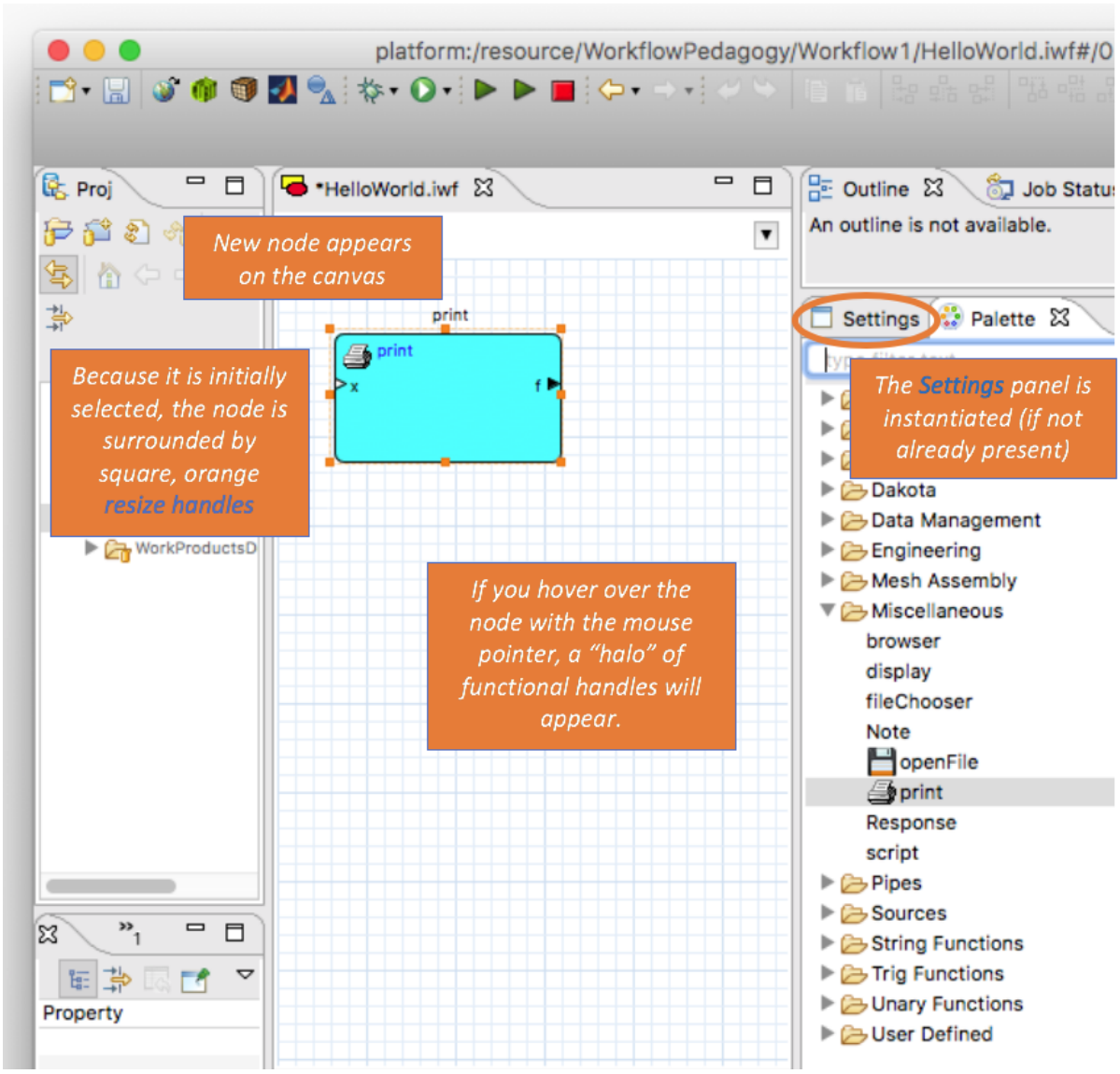
Adding nodes to a workflow

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.



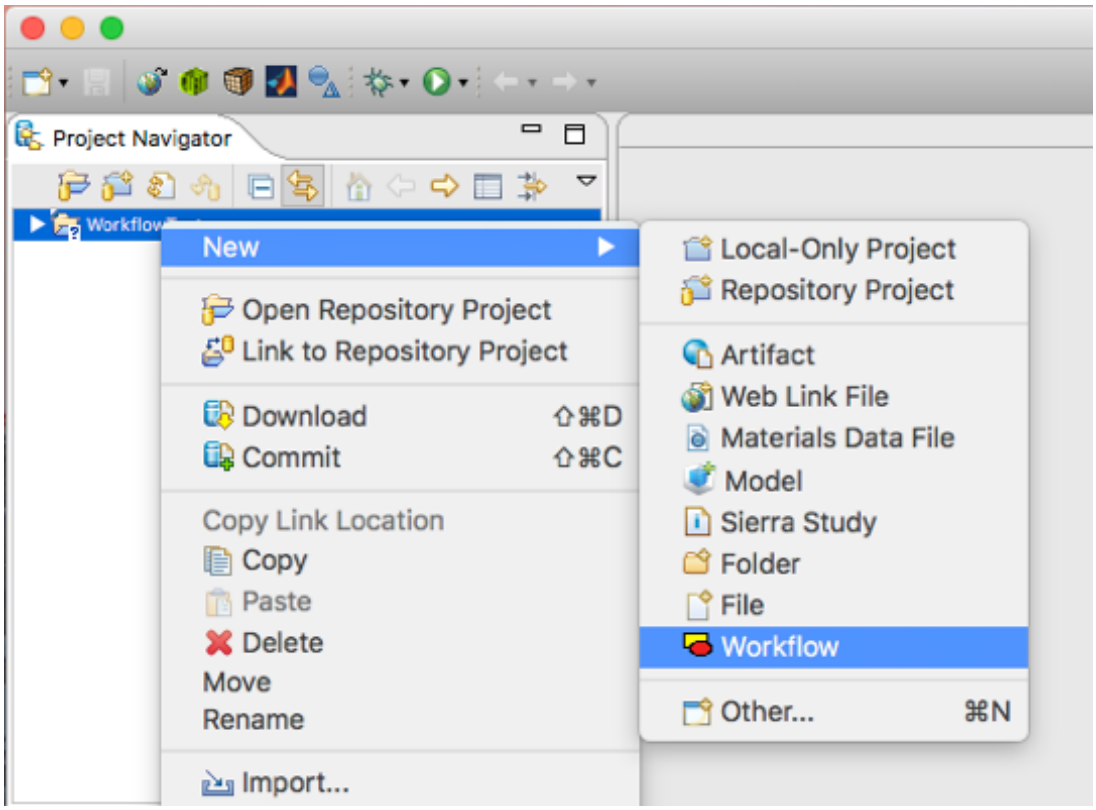
What should happen


platform:/resource/WorkflowPedagogy/Workflow1/HelloWorld.iwf#/0
Proj
*HelloWorld.iwf
Outline
Job Status
An outline is not available.
New node appears on the canvas
print
x
f
Because it is initially selected, the node is surrounded by square, orange resize handles
Settings
Palette
The Settings panel is instantiated (if not already present)
Dakota
Data Management
Engineering
Mesh Assembly
Miscellaneous
browser
display
fileChooser
Note
openFile
print
Response
script
Pipes
Sources
String Functions
Trig Functions
Unary Functions
User Defined
If you hover over the node with the mouse pointer, a "halo" of functional handles will appear.
Property

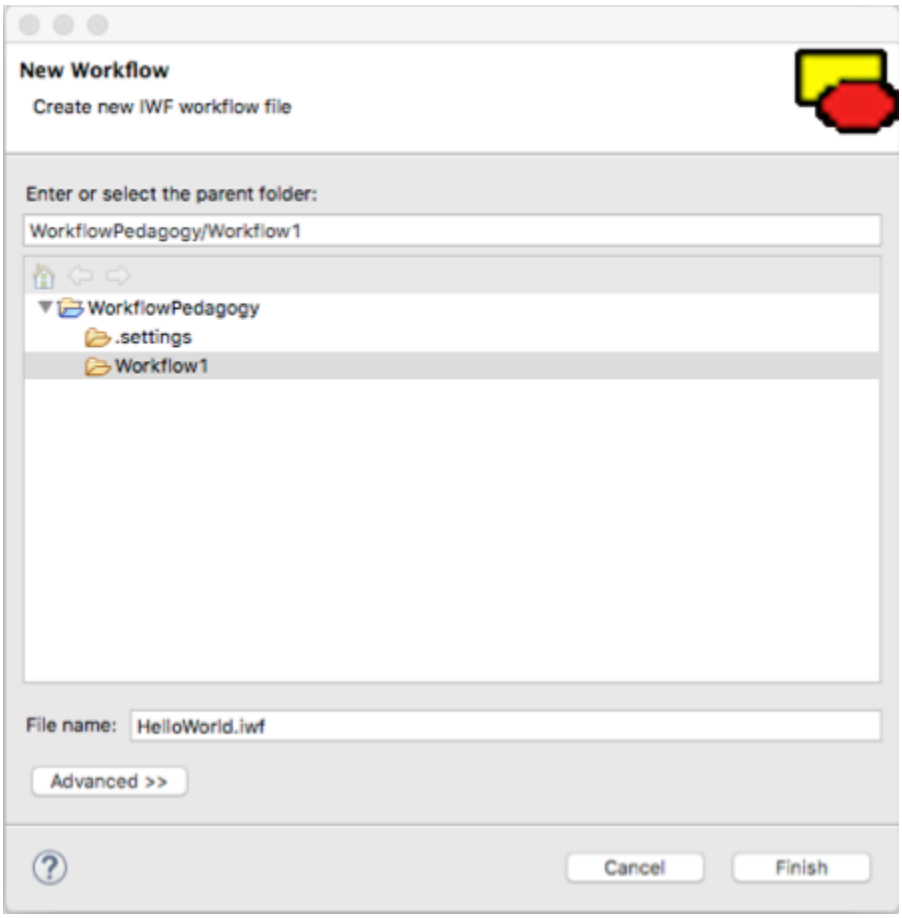
Creating a workflow

What to do:

1. Choose File New Workflow

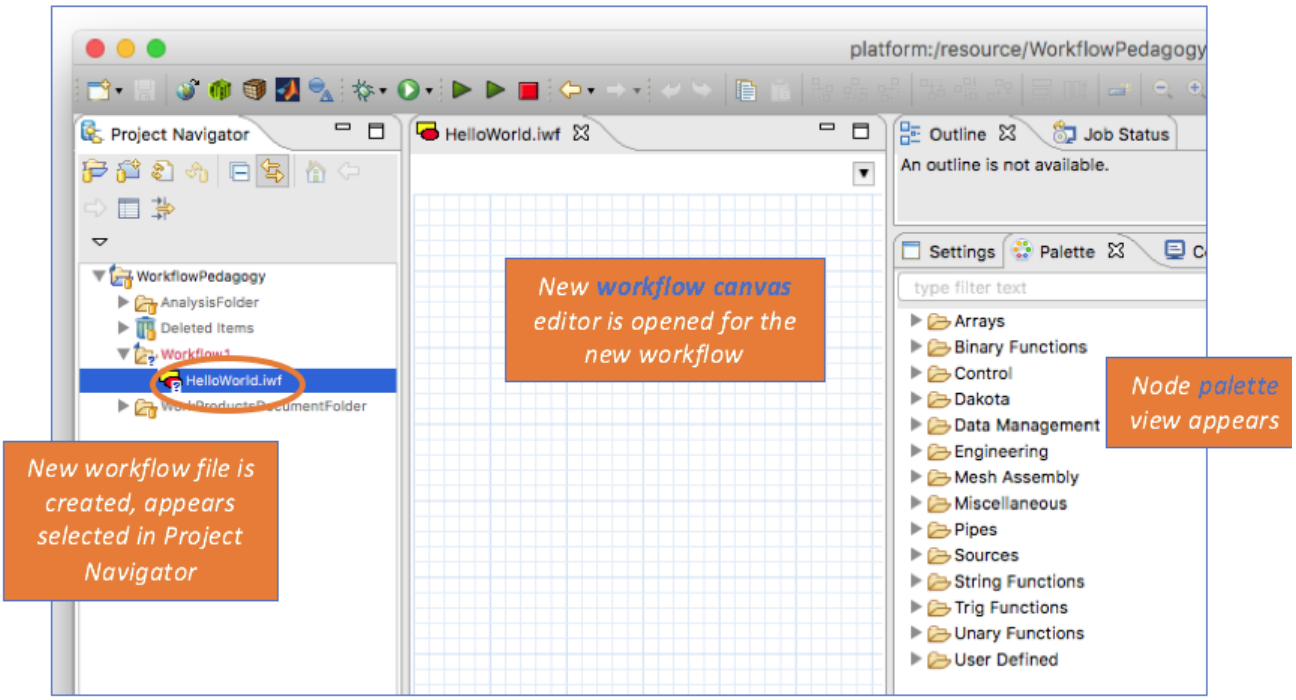


2. Designate the parent folder for the workflow (if one is not already selected), and enter a name for the new workflow file (a `.iwf` suffix will be appended).



3. Click on "Finish"

What happens:



Workflow "Hello, World" tutorial

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

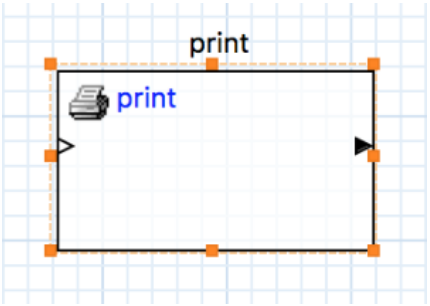
Steps

1. Create a new workflow

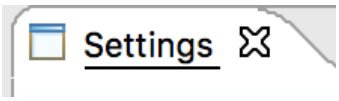
2. Add a **print** node to your workflow

3. Configure the **print** node

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

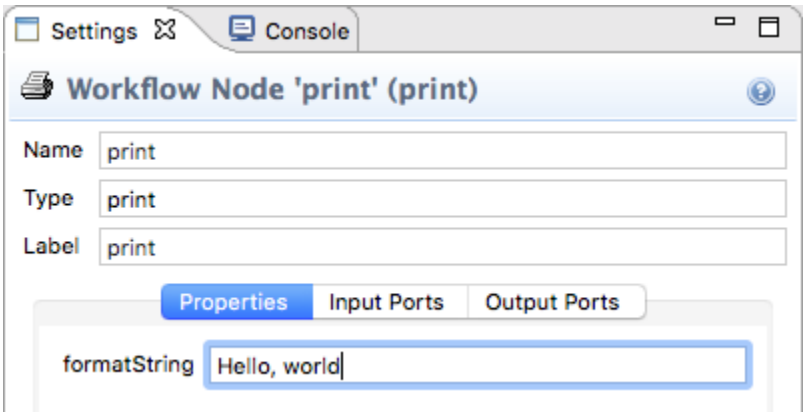
    

    b. Find the tab for the **Settings** panel and click on it to bring it to the foreground.

    (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.

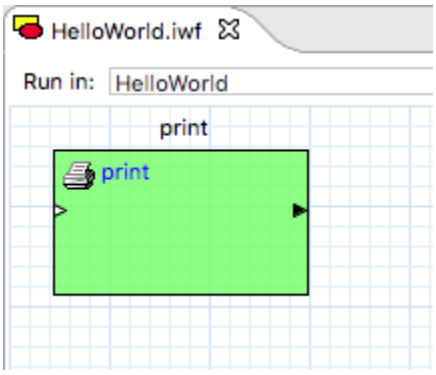
    

4. Execute the workflow from within SAW.

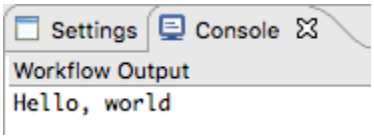
    a. Click on the button in the toolbar near the top of the SAW window to run the workflow.

    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):

    a. Change your working directory to the one where the .iwf file resides.

    b. Create a sub-directory that will be the work directory for this execution of the workflow, and `cd` into this subdirectory.

    c. 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.