
FORCE Workflow Manager

Release 0.5.0

unknown

Sep 28, 2020

CONTENTS

1	FORCE Workflow Manager	1
2	Installation	3
3	Documentation	5
4	User Manual	7
5	API Reference	15

FORCE WORKFLOW MANAGER

This repository contains the implementation of the Workflow Manager UI for the Business Decision Support System (BDSS). It is implemented under the Formulations and Computational Engineering (FORCE) project within Horizon 2020 ([NMBP-23-2016/721027](#)).

INSTALLATION

For full installation instructions and developer documentation, please see the [FORCE BDSS ReadTheDocs](#) page.

DOCUMENTATION

Full user documentation is being hosted at the [FORCE WfManager ReadTheDocs](#) page.

Alternatively, to build the documentation locally in the `doc/build` directory, run:

```
python -m ci docs
```


USER MANUAL

4.1 Introduction

Force Workflow Manager implements a graphical user interface to drive the creation of workflows for the Business Decision System part of the FORCE project. The UI is responsible for creating a workflow file that is then submitted for execution by a command line interface utility, `force_bdss`.

The UI presents a tree view where the workflow is created. The user must specify

- The Multi Criteria Optimizer to use, and the parameter types it should handle for our investigation.
- the Data Sources to use during the evaluation. Data Sources are entities responsible for extracting, computing or generating specific values specific to our computational investigation.

To define each of the above entities, right click on the appropriate entries in the tree. A dialog will open, providing the available entities that can be added, together with entity specific configuration parameters.

Once defined the workflow, it can be saved for later execution (via “Save...”), or executed (by using “Run workflow...”). A previously saved workflow can be reloaded using “Load...”.

During the execution of the `force_bdss`, the results are displayed in the table located in the main pane of the GUI. Those results are also displayed in the plot, the user can choose what results to plot and the clicked point in the plot is synchronised with the selected row in the table.

The execution of the `force_bdss` can be by paused (via “Pause”) and resumed (via the same UI element but with the “Resume” tooltip). The execution can be stopped (via “Stop”), and then `force_bdss` process is terminated.

4.2 Workflow manager concept

As part of task 3.4 we need to develop a workflow manager. The program must be similar to modeFRONTIER, in the sense that not only it makes possible to construct and execute the workflow (constructed from a set of data sources, execution engines, and Key Property Indicators (KPI) extractors), but also to collect the resulting data and perform analysis.

Overall requirements:

1. Ability to compose a workflow “connecting icons” and store it as part of the project. For simplicity, only one workflow per project is allowed.
2. Ability to run the workflow in a queued scheduler, so that the application does not need to run to retrieve the result
3. The executing workflow must dump to a python file which actually performs the execution. We do not expect reading back this file to be feasible to reconstruct the workflow graphically, so we need a more reliable representation for stored workflows.

- once the evaluation is over, the KPI are collected and showed as a spreadsheet and as a plot.

4.2.1 Main window

Opening the program will open the project and thus the associated workflow. The main window should have multiple tabs,

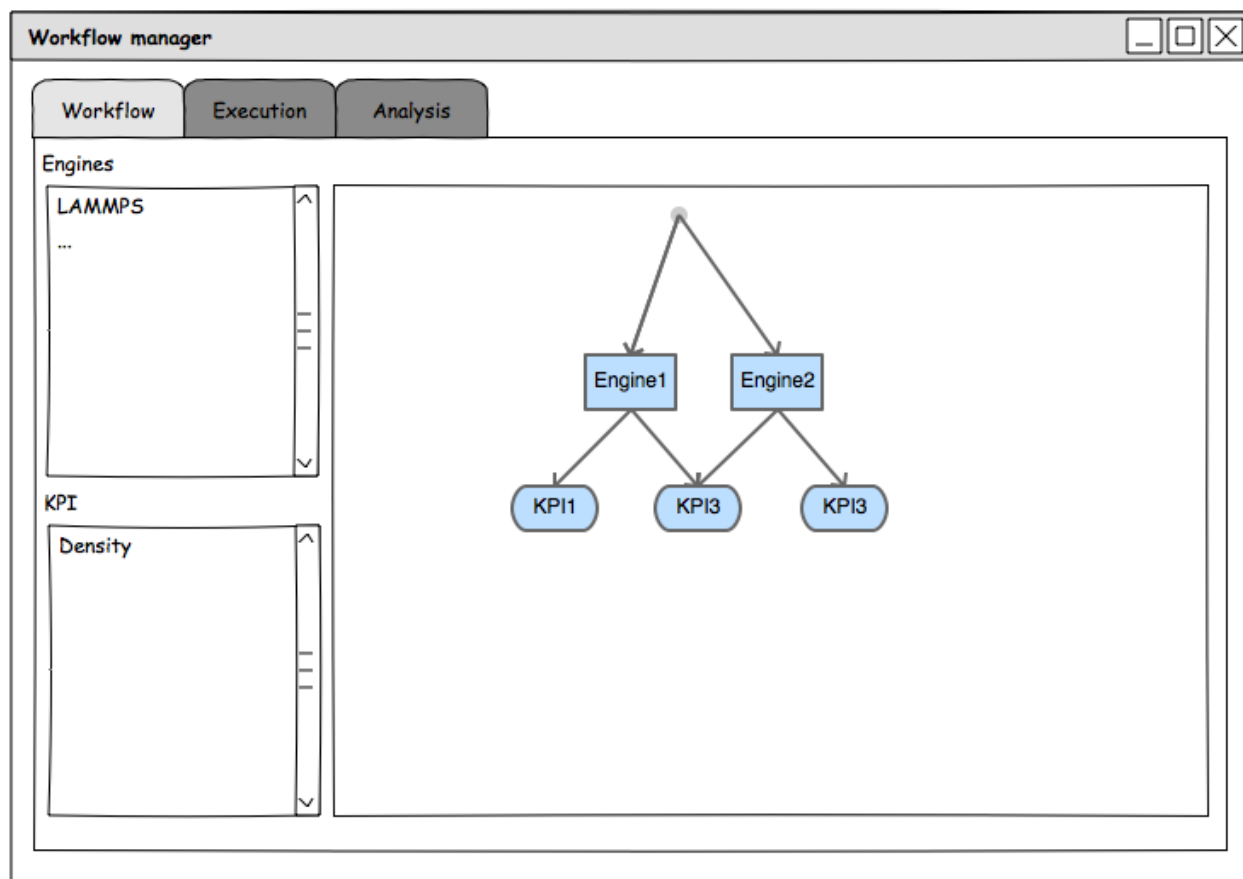
- first tab will show and allow the configuration of the workflow
- second tab has the execution environment, the outputs, the progress and the setup of the MCO.
- third tab has data visualization.

4.2.2 Workflow configuration

The tab content will be split into two parts:

- A palette of available engines, data sources, variables, that can be dragged and dropped on the canvas
- A canvas where the graph from the above components is constructed.

Each individual component can be double clicked to expose its configuration options. A dialog with the appropriate information will be shown.



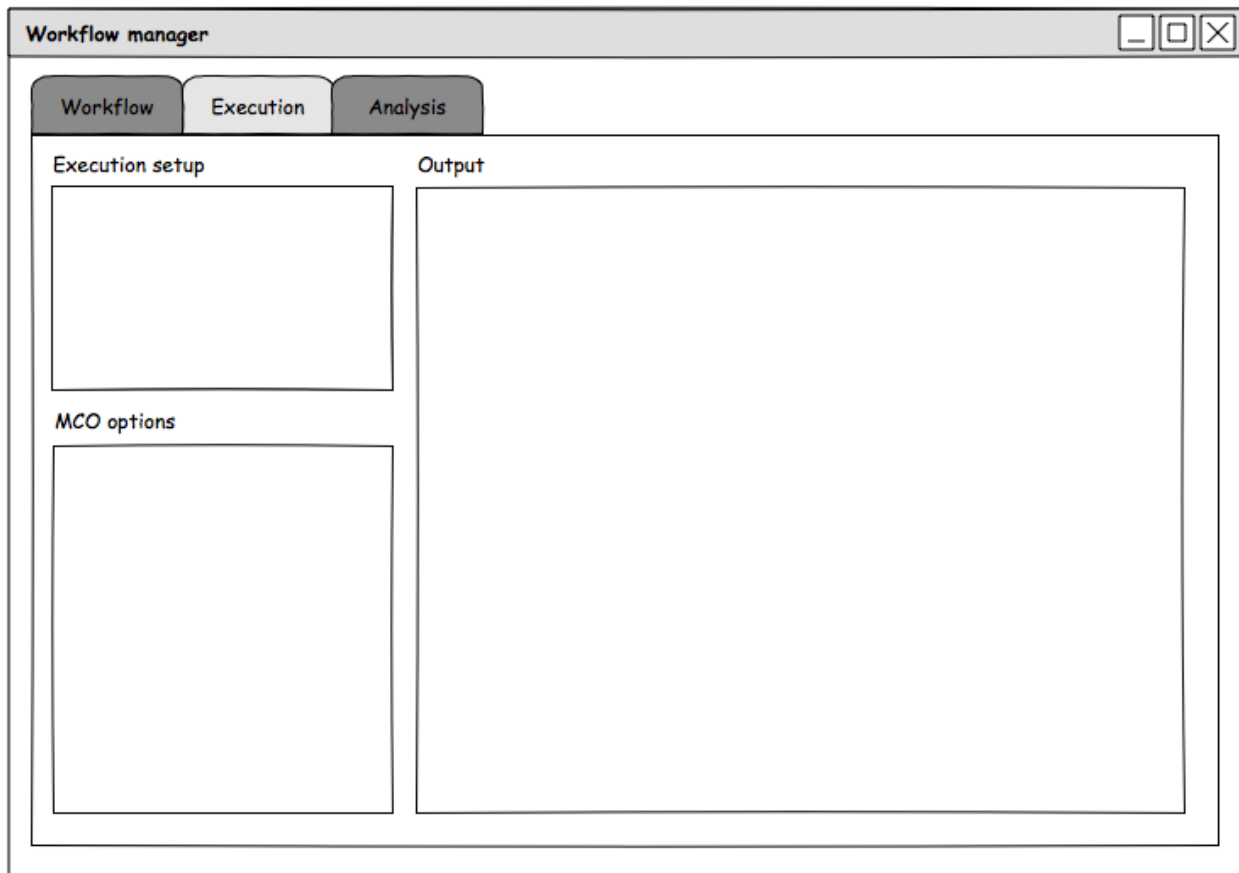
4.2.3 Execution

Once the workflow is set, the execution tab can be used. Once again, the tab content is split into two parts. The first part is configuration for:

- configuration option for the pipeline manager (luigi) e.g. on which machines to dispatch, how many processors, etc.
- Multi Criteria Optimization (MCO) settings, specifically which input variables to modify, the ranges and constraints of each values, which parameter space exploration strategy we want (simplex/genetic algorithm) and which key indicators the MCO should use to evaluate the quality of the result.

Once again, the available MCO come from a palette that is plugin based.

The second part shows the outputs of the various executions.



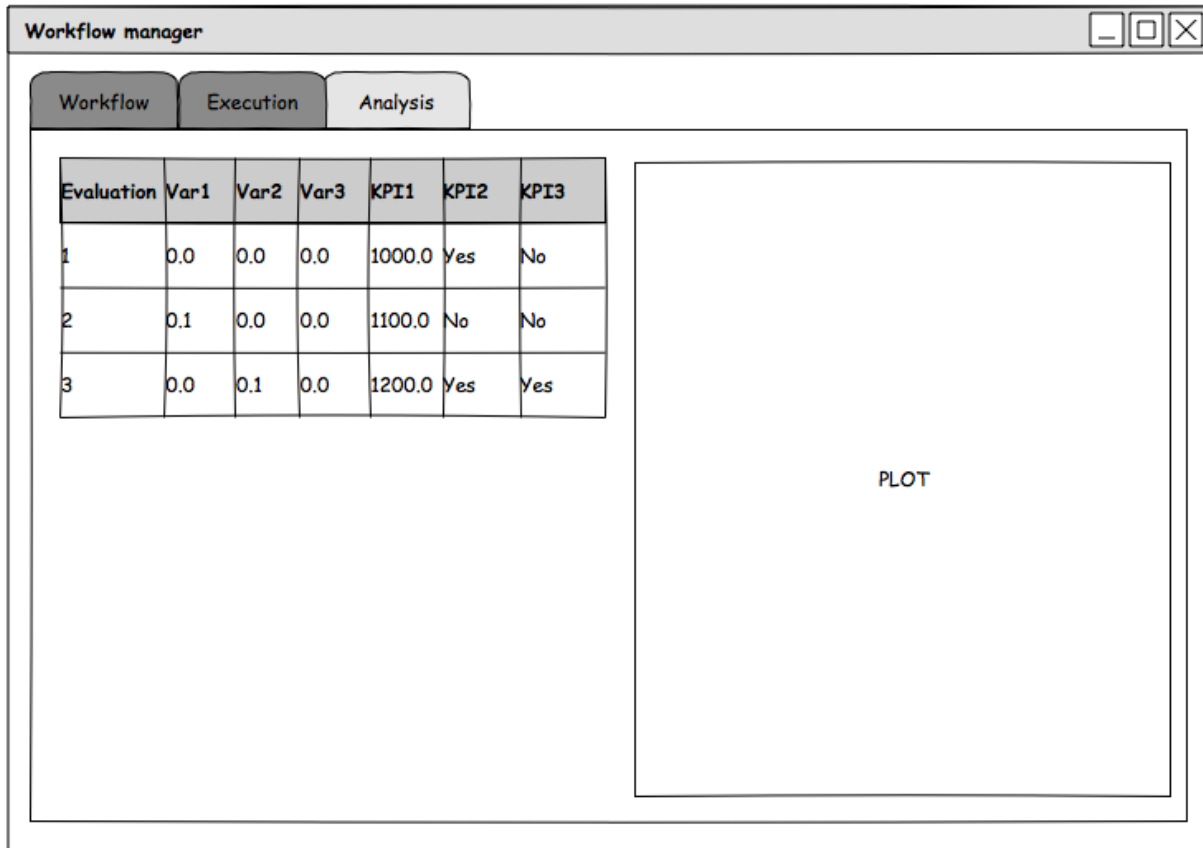
When everything is set, we can start the pipeline, which should keep running regardless if the workflow manager is running or not, meaning that can operate in “disconnected mode” and reattach at a later stage when restarted.

To manage the pipeline, we plan to use Luigi. The workflow graph will be rendered into a python script whose content appropriately map the graphically expressed workflow. Luigi will drive this pipeline in the script.

It is at this stage unclear how the script can retrieve the state of the pipeline as it progresses, but we can take advantage of Luigi web client, maybe?

4.2.4 Analysis

As the various runs from the MCO complete, the KPIs are extracted. The associated input data and the KPIs are shown in the third tab, which provides both a tabular view (excel-like) and a plot view (with chaco) of the points.



4.2.5 Design notes

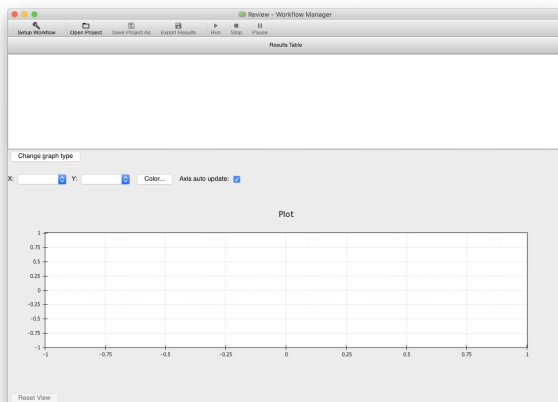
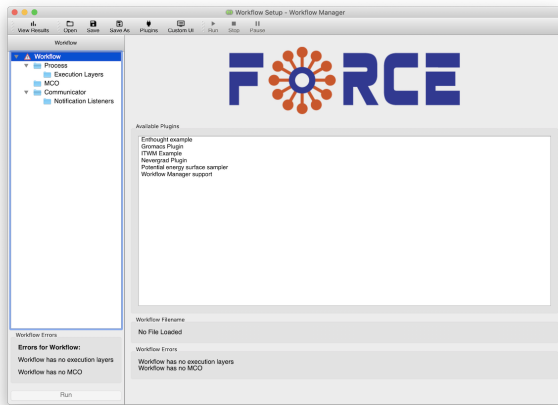
Clearly, the MCO must direct the execution and the creation of new input according to the obtained results, so they must be in an orchestration role, rather than being part of the workflow. In other words, the workflow is a script that runs independently, and the MCO decides the input to pass to it and thus executes as part of our workflow manager.

MCO, KPI evaluators, and Engines, must export information not only for the integration in the framework, but also to present UI elements (config dialogs, icons) to the workflow manager UI.

Having the MCO separated from the pipeline means that our python script with the pipeline can run only on one set of parameters, and the MCO is external. Do we want the script to be able to do MCO as well, thus delivering the whole KPI table once run?

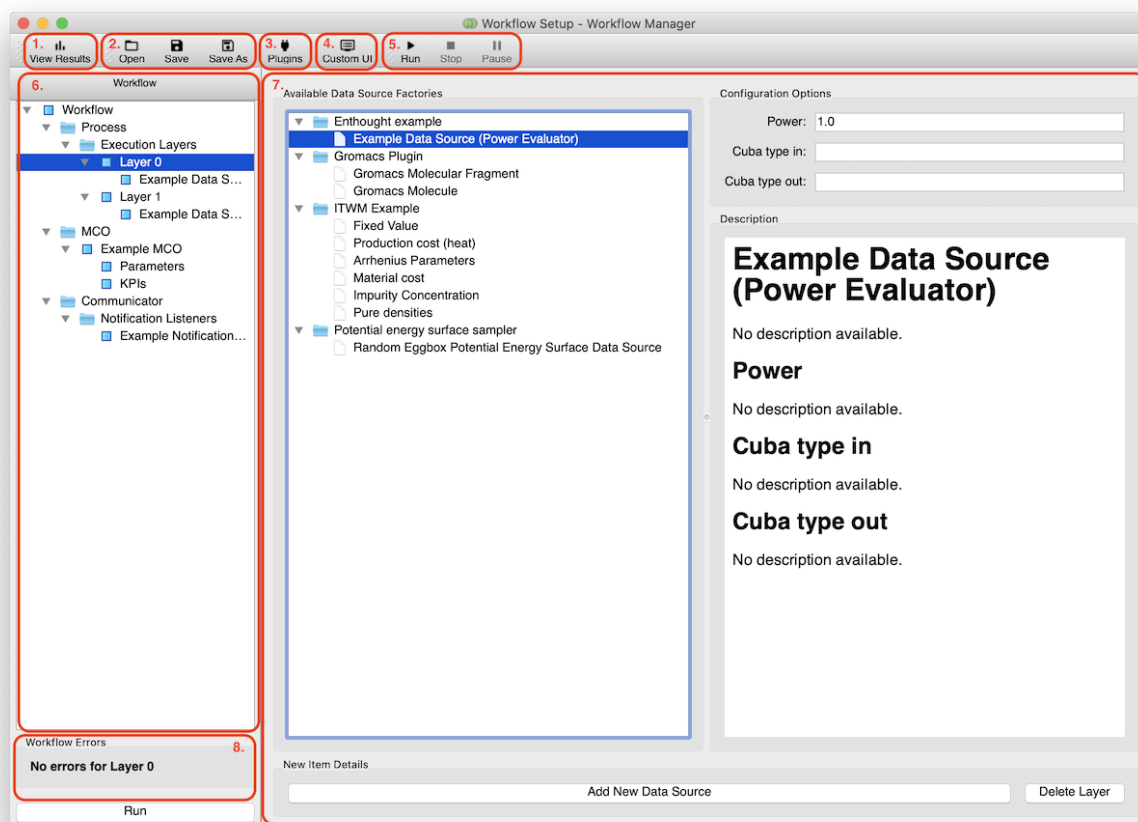
4.3 WfManager GUI Design

The FORCE WfManager is built on top of the [Envisage](#) framework for extensible GUI applications. As such, at the heart of the application is the WfManager class, which inherits from `envisage.TaskApplication`. The current design of the FORCE WfManager contains two `pyface.Task` implementations: the `WfManagerSetupTask` and `WfManagerReviewTask` classes. These objects provide the functionality to construct workflows and analyse the results of an MCO respectively. Switching between tasks is performed by using the “Setup” and “Review” toolbar icons.



4.3.1 Setup Task

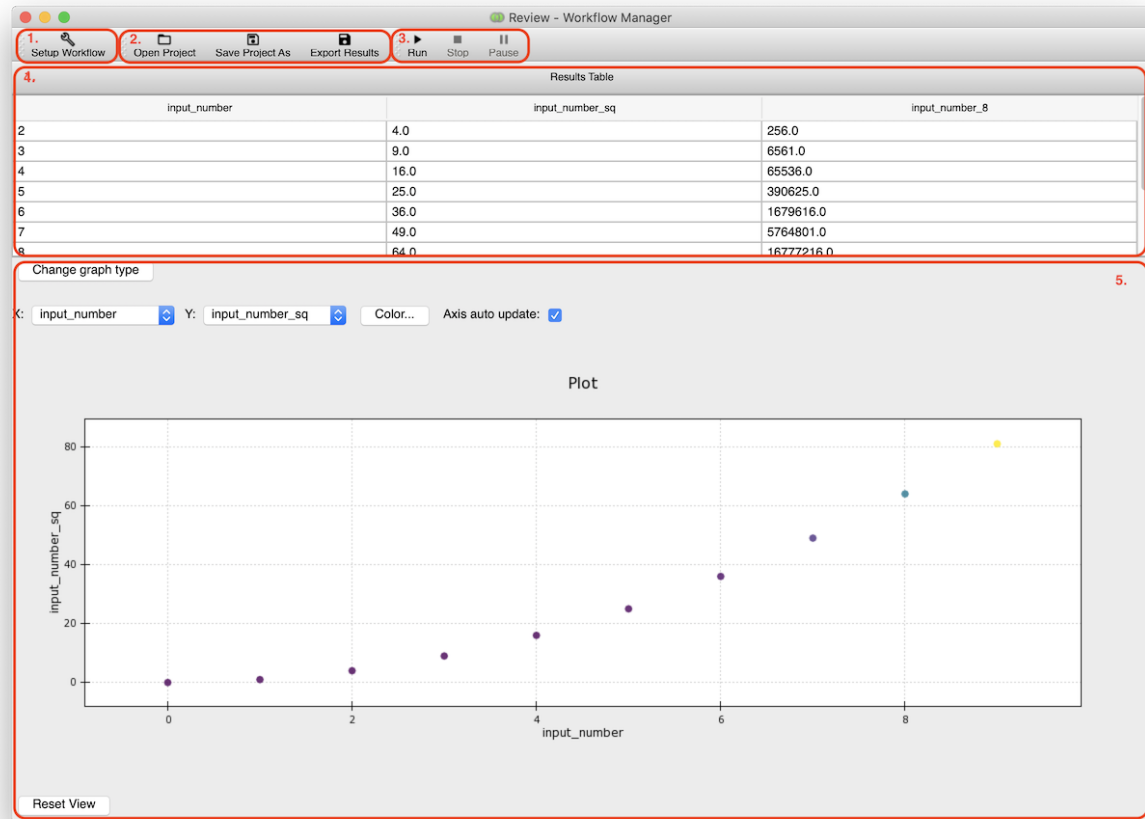
The Setup task is used to construct workflows to optimise in the BDSS. As well as providing UI features to edit model parameters related to each component in the workflow (data sources, notification listeners, MCO etc.) it also allows the user to define how variables are passed between input and output slots between execution layers.



Number	Description
1. View Results	Review task icon, switches view to the Review task
2. Workflow IO	Shortcut icons for opening / closing Workflow files
3. Plugins info	Brings up plugins window, including a description of each BDSS plugin detected
4. Custom UI window	Brings up custom UI selection window, used to start
5. MCO control	Start / stop / pause controls for the MCO
6. Workflow tree	Tree selection panel for parts of the Workflow under construction
7. Workflow editor	View used to edit selected parts of the Workflow under construction
8. Workflow Errors	Displays errors reported by verification routine of Workflow under construction

4.3.2 Review Task

The Review task is used both display the results and progress of the MCO being performed on a constructed workflow. It contains a table of values that are reported back to the WfManager during an optimization procedure, as well as a selection of 2D plots to view these values.



Number	Description
1. Setup Workflow	Setup task icon, switches view to the Setup task
2. Project IO	Shortcut icons for opening / closing Project files
3. MCO control	Start / stop / pause controls for the MCO
4. MCO results table	Table displaying data reported back from an MCO, with each row representing an optimal point identified
5. MCO results plot	Plot displaying data reported back from an MCO

4.4 License

Copyright (c) 2017, Enthought Ltd All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The icons are mostly derived work from other icons. As such they are licensed accordingly to the original license:

4.4.1 Project License File

Enthought BSD 3-Clause LICENSE.txt

Unless stated in this file, icons are the work of Enthought, and are released under a 3 clause BSD license.

4.4.2 Files and original authors:

force-wfmanager/force_wfmanager/left_side_pane/icons: all | Enthought

API REFERENCE