

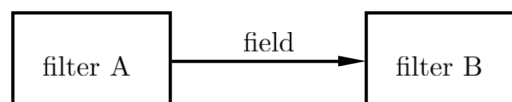
User Guide for XIOS workflow Graph

Yushan Wang

October 14, 2019

1 What is the workflow graph

The workflow graph generated by XIOS shows the path of the data flux in the XIOS context. It can give user an overall presentation of how the field is generated and which operations are applied. It is often not easy to track the data dependency from the `xml` file. That's why a graphic representation is more user-friendly.



1.1 Node

In the workflow graph, a node represents a XIOS filter. We have 7 categories of filters:

- source filter: when the field data is received from model or read from netcdf file
- pass through filter: when having a field reference
- arithmetic filter: when arithmetic operations are performed on the field
- temporal filter: when the field has a temporal operation (*e.g.* average, maximum, minimum, ...)
- spatial transform filter: when the field has a spatial transform applied upon (*e.g.* zoom, interpolate, ...)

- file write filter: when the field data is written to a netcdf file
- store filter: when the field data is send back to model

1.2 Edge

Edge in the workflow graph represents the field after the filter is applied.

2 Enable the workflow graph

The field attribute `build_workflow_graph` is used to control the workflow graph.

```
<field id="field" grid_ref="grid" build_workflow_graph="true" />
```

2.1 Heritage by field referecing

```
<field id="field_A" grid="grid" build_workflow_graph="true" />
<field id="field_B" field_ref="field_A" />
```

2.2 Heritage by arithmetic operations

```
<field id="field_A" grid="grid" build_workflow_graph="true" />
<field id="field_B" field_ref="field_A" > field_A*2 </field>
```

3 Visualization of workflow graph

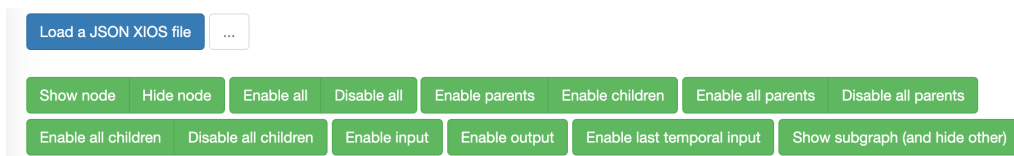


Figure 1: Start state of the Visualization

The workflow graph can be viewed through the `graph.html` script located in the XIOS folder. The graph file `graph_data.json` is stored in the same folder as the `iodef.xml` file which is used for configure the simulation. Choose the graph file to start the Visualization by clicking the button `Load a JSON XIOS file`. Several buttons are set to allow user to visualize the graph interactively.

3.1 Enable/Disable all

The graph is hidden at the beginning. `Enable all` will show the whole graph and `Disable all` is used to hide the whole graph

3.2 Show/Hide node

The **Hide node** buttons allow user to temporarily hide a node and its edges. The node's position is kept and user can still click on it to show the node and its edges by the **Show node** button.

3.3 Enable parents/children

These buttons can show the direct parents or children of a given node.

3.4 Enable/Disable all parents

These buttons are used to show/hide all parent nodes and edges of a given node.

3.5 Enable/Disable all children

These buttons are used to show/hide all child nodes and edges of a given node.

3.6 Enable input

Enable input shows all input nodes of the graph. The input nodes are often, if not always, the source filters: data received from the model or read from a netcdf file.

3.7 Enable output

Enable output shows all output nodes of the graph. The output nodes are either the file writer filter: data written to a netcdf file; either the store filter: data send back to model.

3.8 Enable last temporal input

Very often, a field is generated after many iterations. Showing all these iterations can be redundant and slow down the browser. That's why the **Enable last temporal input** button is useful because it can show us only the last iteration before the flux gets to the current filter node.

3.9 Show subgraph

Show subgraph button is very useful because it can focus on a given node and show all its parents and children nodes and edges, and at the same time hide all irrelevant elements to the given node.

4 Hover message in the workflow graph

A hover message is shown for each filter and field. For example, the hover message for a file writer filter gives all attributes of the output file and the written field.

5 Time control of workflow graph

Attributes `build_workflow_graph_start` and `build_workflow_graph_end` are used to control the starting and end time of the workflow graph.

```
<field id="field" grid_ref="grid" build_workflow_graph="true"
      build_workflow_graph_begin="3ts"
      build_workflow_graph_end="5ts" />
```

In this example, the graph will only include information between the 3rd and 5th timestep. These attributes are not mandatory. If `build_workflow_graph_start` is not defined, the graph will start to record from the first iteration. Similarly, the default value of `build_workflow_graph_end` is the last iteration of the simulation.

In case that two values of the same attribute are defined, we take the largest range.

```
<field id="field_A" build_workflow_graph="true"
      build_workflow_graph_begin="3ts"
      build_workflow_graph_end="5ts" />
```

```
<field id="field_B" field_ref="field_A" build_workflow_graph="true"
      build_workflow_graph_end="4ts" />
```

In this example, `field_B` will have attributes `build_workflow_graph_start="1ts"` and `build_workflow_graph_end="5ts"`