

Progress in Workflow Comparison

N. Oakes

*CSIRO Data61, Australia
Email: nerolie.oakes@data61.csiro.au*

Abstract: The CSIRO has been developing the collaborative Scientific Workflow System (SWS), Workspace, since 2005. The Workspace Workflow editor is a graphical tool used to both design and edit workflows. Typically, workflow design is an ongoing process. Workflows can have multiple configuration settings and can be saved in an XML format. While XML, as a text-based format, can be incorporated easily into source control utilities, differences between revisions can be difficult to understand by comparing changes in XML elements. Workspace ships with a several executable tools to help minimise this difficulty: the Workspace comparison tool, aimed at helping developers track changes to workflow versions over time; and the global name comparison tool, aimed at helping developers compare configuration parameters between workflows that expose their capabilities via an interface.

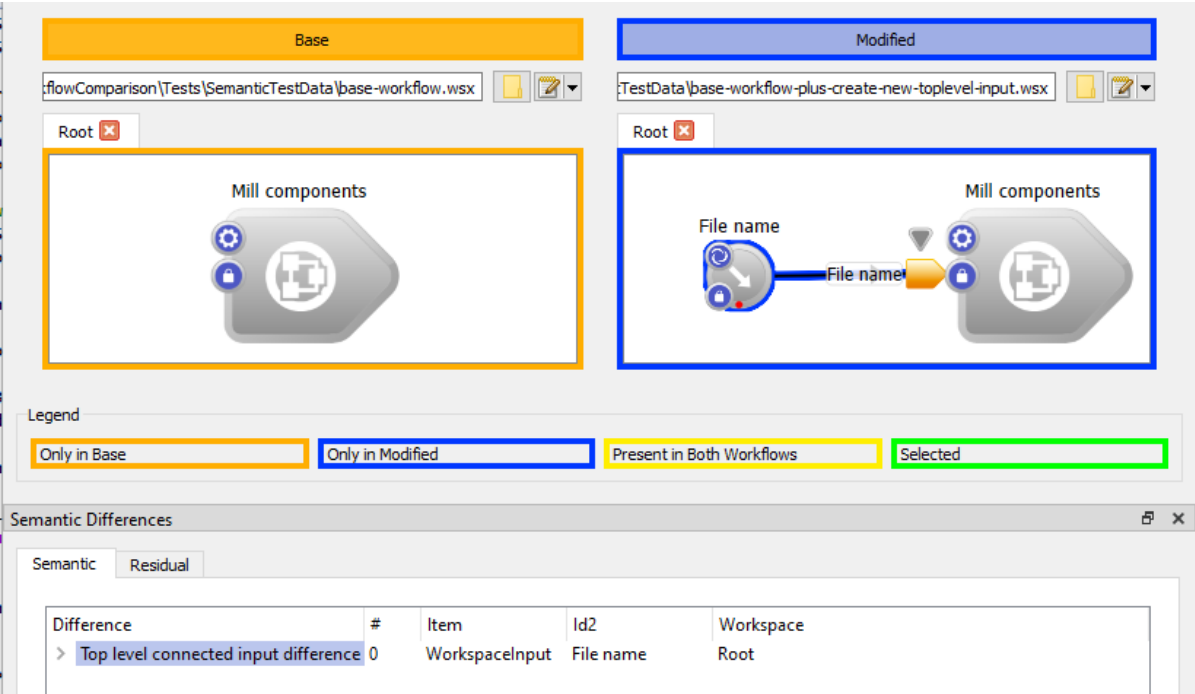


Figure 1. Semantic differencing in the Workspace Workflow Comparison Tool

In this paper we discuss the global name comparison tool, and recent improvements in workflow comparison, most notably, semantic differencing which has been recently introduced to conceptually group related changes to workflows, making it easier to understand differences between workflow versions.

Keywords: *Workspace, workflow, visualisation, XML Comparison*

1. INTRODUCTION

The CSIRO's Scientific Workflow System (SWS), Workspace, was first released publicly in 2014. It can be used free of charge and consists of a suite of development tools, the most important being the Workspace editor – a graphical tool that can be used to both develop and run Workflows. The suite is under continual development with four driving themes: Analyse, Collaborate, Commercialise and Everywhere (Bolger, M. *et al.* 2016 ;Watkins et al 2017; Oakes et al 2019).

A Workspace workflow (Fig 2) is substantially a set of connected operations, where each operation represents a task, which can be visualised and modified via the Workspace editor. A workflow is comprised of both visual and functional components – visual components are elements such as layout, labels, and display widgets while functional components are operation types, input names and values, as well as “global names”, which expose operation inputs/outputs for use via an interface such as a custom application or command line.

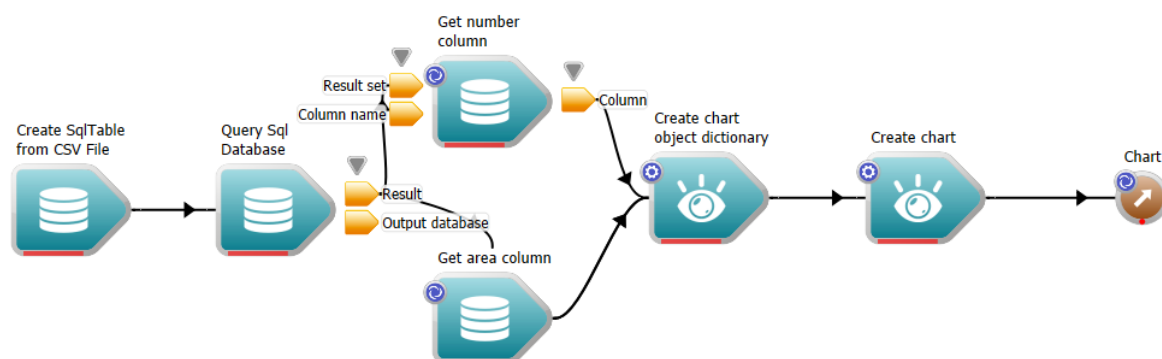


Figure 2. A Workspace workflow contains both visual and functional elements

Workspace workflows can themselves contain “nested” workflows, where a group of operations are treated as a single operation, making the workflow easier to follow.

2. WORKSPACE COMPARISON TOOLS

Two of the tools supplied with the Workspace installation are: the Workspace workflow comparison tool, used to compare different versions of a Workspace workflow; and the global name comparison tool, used to compare two groups of global name settings. These can be two sets of settings for the same workflow, or for different versions of the workflow. The Workspace comparison tool (described in Oakes, N et al, 2021) is an XML comparison application specifically designed to visually show the differences between Workflow versions both as a side-by-side set of interactive Workflow visualisation panels (Figure 3) and an ordered list of low-level differences.

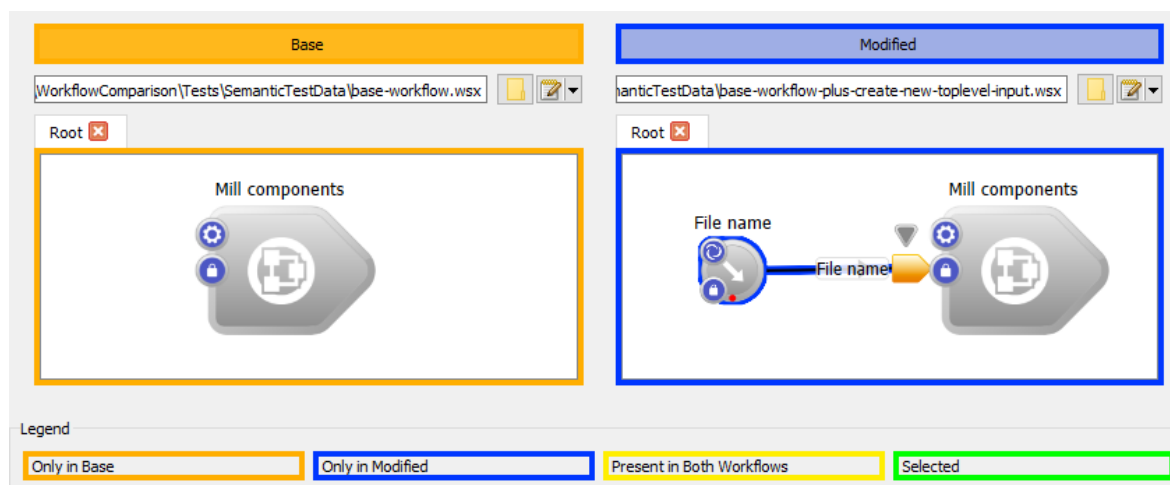


Figure 3. The Workspace diff tool graphical "Workflow" panel where the base workflow is on the left and the modified workflow on the right

The modified version of the simple workflow shown in Figure 3 was created from a single user menu click (to add a way of setting a filename used by an operation within the “Mill Components” nested workflow). While the graphical panel shown makes it somewhat clear what the difference is, there are more than 20 differences between the two XML files. In this paper, we describe how the comparison tool works to understand changes on a higher level by grouping together linked changes to the XML files (ideally, this single user action would be identified as one, rather than twenty individual differences – see Figure 4). This is described in the section Semantic Differencing.

Semantic				
Residual				
Difference	#	Item	Id2	Workspace
> Top level connected input difference	0	WorkspaceInput	File name	Root

Figure 4. A difference identified as the single user action of adding a top-level input

The second part of the paper will give a brief description of the Workspace global name comparison tool.

3. SEMANTIC DIFFERENCING

The development of the Workflow comparison tool has been guided by a few principles: 1) the two workflows are assumed to have a common ancestry, 2) it should be easy to identify and understand differences, and 3) it should highlight the most relevant differences between the two workflows. To satisfy principles (2) and (3), we cannot simply highlight a list of low-level differences to the user. Instead, we must present the information in a way that maps to the conceptual ‘actions’ that were undertaken by the user when modifying the workflow. The tool displays this information in two different ways. The first is to display a pair of workflows, familiar to users accustomed to designing their applications using the Workspace editor (see Figure 3). The second is to display a tree of differences showing a detailed comparison of differences between the two XML files (see Figure 6). Both have advantages and disadvantages.

Let us take as an example the workflows shown in Figure 3 created by a user wanting to be able to set a parameter needed deep in a nest from the top level of the workflow. To create this difference in the Workspace editor, the user needed to perform just a single action (one click) (see the Workspace manual, 2023):

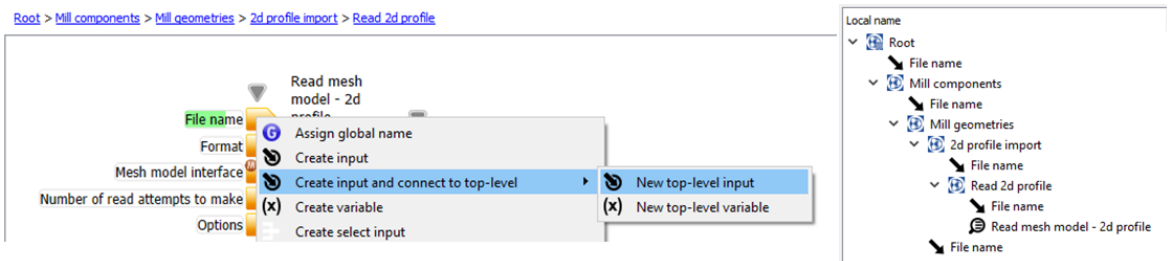
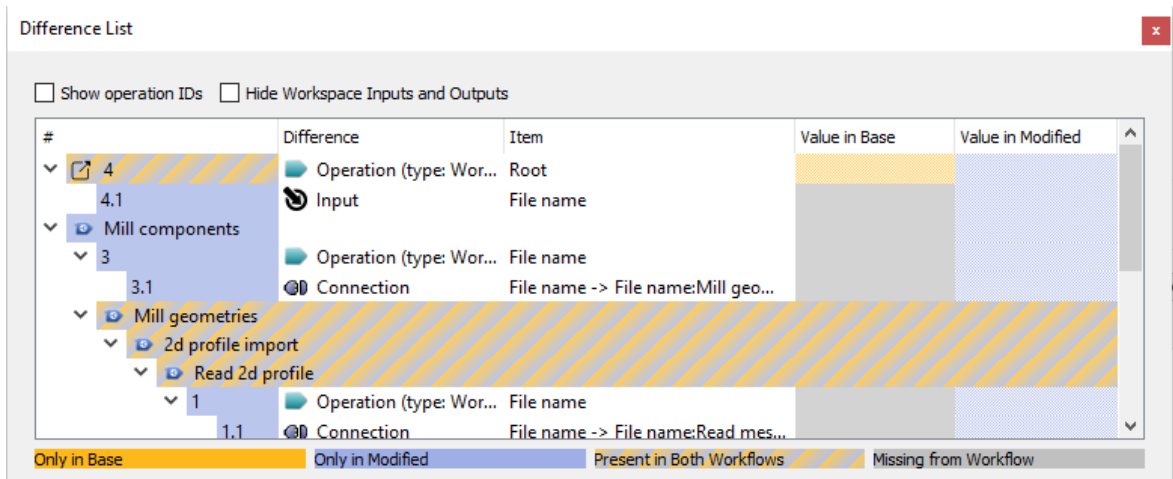


Figure 5. A user can create a top-level input from a deeply nested operation with single menu item

The operation that needs a value for the File name is “Read mesh model – 2d profile” and it is inside the nested workflow Read 2d Profile, which is inside 2d profile import and so on. Inside each nested workflow a new input operation is created together with its connections. The resultant chain of new input-operations is shown in the workflow tree on the right-hand-side of Figure 5.

The result in the graphical panel is shown in Figure 3 – it is clear there is a new top-level input, but it is not clear to which operation and nested workflow it belongs. The result in the text panel is shown in Figure 6, here it is much easier to see everything that has changed, but it has picked up over twenty individual differences that were created by this one action!



#	Difference	Item	Value in Base	Value in Modified
4	Operation (type: Wor...	Root		
4.1	Input	File name		
3	Operation (type: Wor...	File name		
3.1	Connection	File name -> File name:Mill geo...		
Mill geometries				
2d profile import				
Read 2d profile				
1.1	Connection	File name -> File name:Read mes...		

Only in Base Only in Modified Present in Both Workflows Missing from Workflow

Figure 6. Adding a top-level input, Difference List Window

In this example, while not ideal, with only one difference it would not be difficult to understand the intention of the changes. However, it's likely in practice that a number of inputs would be created at the same time. If there were just five new inputs added, resulting in one hundred individual differences, it quickly starts to get confusing. With this in mind, the research team has been working on identifying ‘semantic’ changes: those that more closely represent the high-level changes the user made to the workflow.

3.1. A new differencing workflow

The workflow comparison tool is itself built on a Workspace workflow, and Figures 7 and 8 show the changes to the workflow both at a root and inside the nested workflow “Identify Semantic Differences”. Essentially, the low-level differences are calculated by comparing the two XML files as usual, the “Identify Semantic Differences” operation is run over those to identify semantic differences, these are then merged back into the base workflow so the differences that have not been accounted for can be shown in the “Residual” dock.

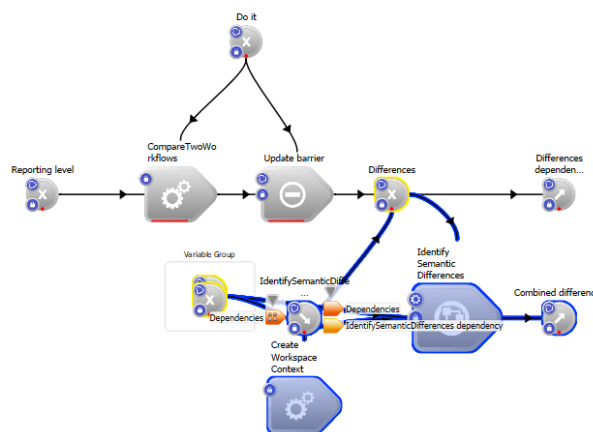


Figure 7. The Semantic Differencing workflow runs after the low-level differences have been identified

By analysing the actual changes made between versions saved to a shared source code repository, we can ascertain that the most common element changed is connections, followed by operations. We can also find that in most instances where a connection is added or removed, it is done so along with an operation. By detecting pairs of new (or deleted) operations and connections, and by linking multiple pairs of operations from deeply nested workflows to the root, we can greatly reduce the number of trivial changes we show and make it much easier to understand what has happened. Figure 9 shows how a semantic difference is shown, along with type-specific information. Work is currently underway to extend this robustly to nested and exploded workflows as these are also common changes that involve lots of small and predictable sets of differences.

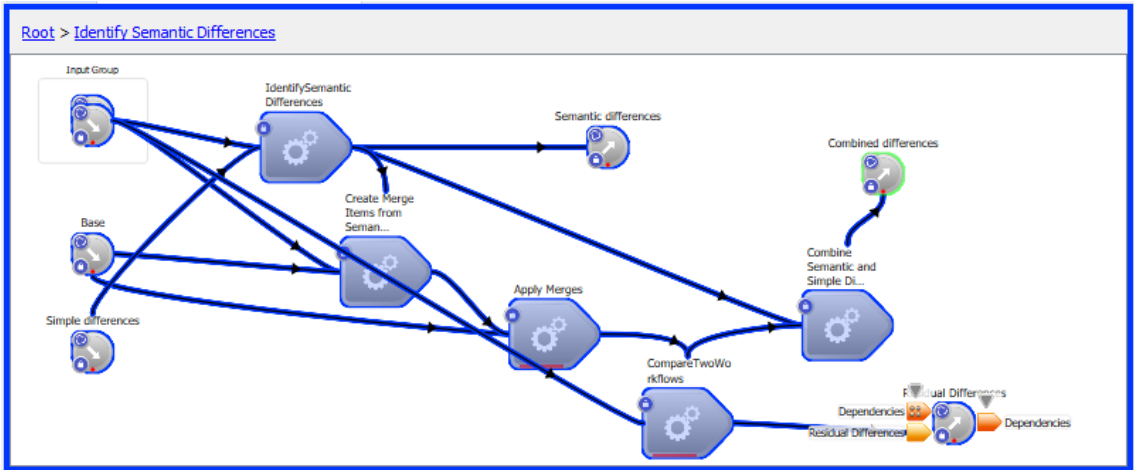


Figure 8. Inside the nested workflow - the identified changes are merged back into the base workflow and residual differences calculated

Semantic Differences				
Semantic Residual				
Difference	#	Item	Id2	Workspace
Top level connected input difference	0	WorkspaceInput	File name	Root
Connected input	0			Mill components
Connected input	1			Mill geometries
Connected input	2			2d profile import
Connected input	3			Read 2d profile

Figure 9. It also has type-specific information

We currently assume that changes were made through simple changes to a base file via the Workspace editor. Future work will involve extending this to comparing two modified versions of a base workflow (perhaps with changes made by different collaborators) and identifying that they made similar changes.

4. GLOBAL NAMES

A “global name” is an identifier given to an operation, input or output that is assumed to be unique over an entire workflow. Consequently, a global name allows a particular input or output to be identified without having to know where it is in the workflow, or even which operation it belongs to.

Global names help to identify key inputs and outputs. As they are globally unique, the values of inputs and outputs with global names can be changed or read using linked external applications. Sometimes developers seek to simplify things for the user, and global names help identify a few key inputs and/or outputs. Importantly, you can use global names to create custom user interfaces, custom widgets and custom applications without having to generate a huge amount of code. Workflows for large projects are often complex and have many different input values that control how they operate. When workflows have many parameters, it can be expedient to have a robust way of saving and restoring parameter values for particular scenarios. This can be done through use of “global name” (settings) files. Global name files are useful in different ways such as: (a) they are a record of the settings used; (b) when the user next needs to run the workflow they can load a settings file to return to a previous known good set of parameters; (c) they can be used as a single parameter if you run a Workflow from the command line, rather than having to set an argument for each parameter; and (d) the user can run the same workflow with different settings, similarly to running a program with different command-line arguments.

When working with the workspace editor, users can edit and keep track of global name settings using the canvas (Figure 10) or the Global Names window (Figure 11).

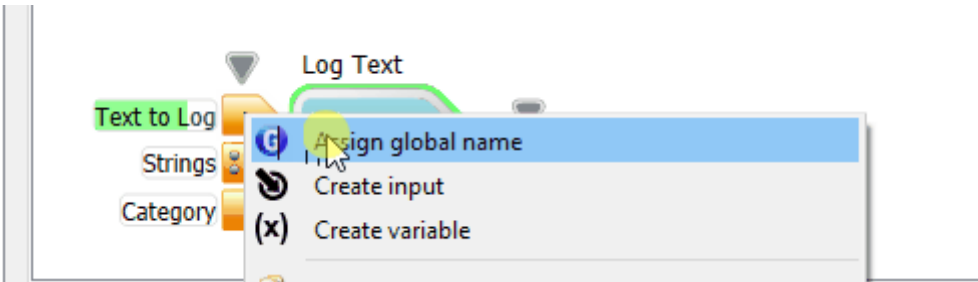


Figure 10. Assigning a global name in the Workspace editor

Global Names						
	Namespace	Global Name	Workflow Item	Data Type	Data Value	Select Display Widget
1	<div></div>	column	Column name	QString	<div>area</div>	<div>QLineEdit</div>
2		chart	Create chart	CSIRO::DataAnalysis::Chart		<div>ChartWidget</div>

Figure 11. The Workspace Editor's Global Names window

5. GLOBAL NAME COMPARISON

The Workspace global name comparison application (Figure 12) can be used to pinpoint any differences or errors between two global name files. You can also use it to see what global names have been set and edit values.

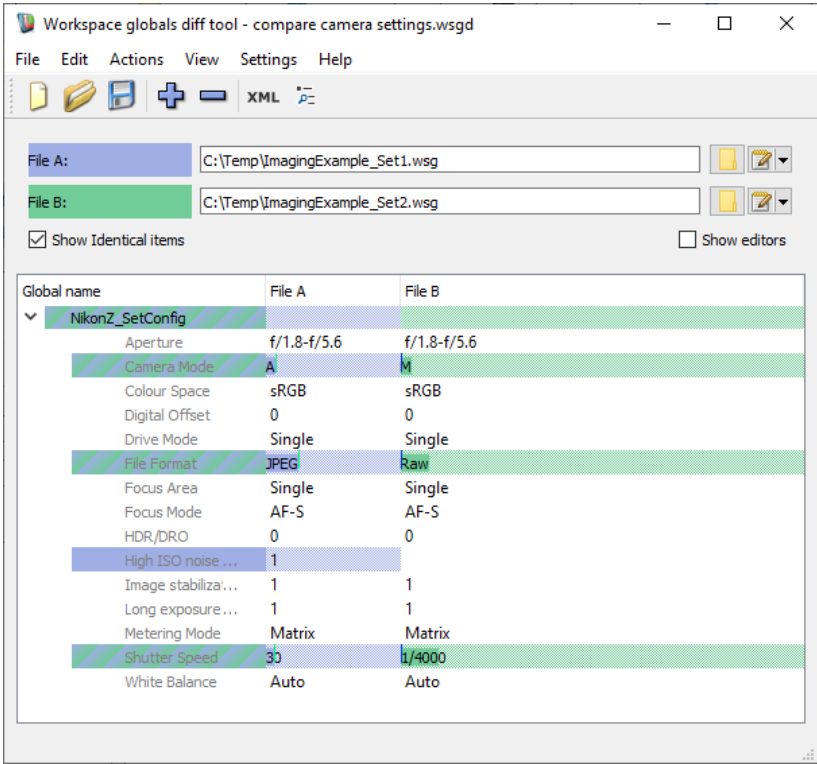


Figure 12. The Workspace global name comparison tool

The application has a single panel that presents all the global names in a tree widget form. It can be used to view the settings of a single file or to compare two related files. Each global name is shown in black font in the “Global name” column, and the values (if they exist) in the “File A” and “File B” columns. Note that a single global name can represent a group of settings (for example, an operation identified by a global name may have multiple input parameters). Such subordinate parameters are identified by names in grey. If the

global name exists in just one file, the cell will be filled in a single colour, where it exists in both and the values differ, the cell will be filled with a diagonal brush, and where the values are identical, they will either not be shown or have a no fill colour (depending on the “Show Identical Items” checkbox state).

This application can be used to edit values. Where the values are not text, the application attempts to identify the data type and supply an appropriate editor (see Figure 13).

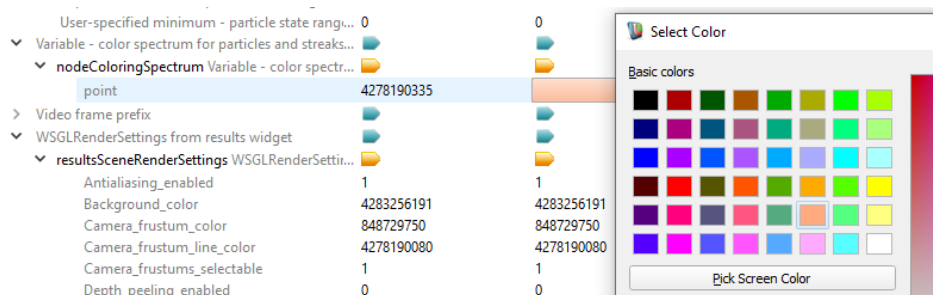


Figure 13. The application tries to supply an appropriate editor. In this case, a colour-selection widget. The user can override this and directly choose the best editing widget by checking the **Show Editors** box (Figure 14).

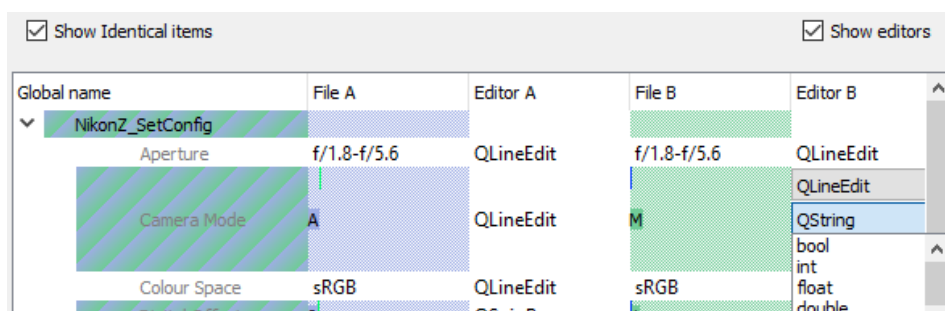


Figure 14. The user can override the default value editor

6. DISCUSSION AND CONCLUSION

The Workspace workflow comparison tools are domain-specific XML comparison tools under ongoing development aimed at making workflow differences simple to understand. While the configuration files are simple and the tool commensurately complete, Workspace workflows can be complex and the resultant XML files have many minor changes which can be difficult to interpret. We are beginning to simplify this process by linking operations and connections, and groups of these with the aim of matching the number of displayed differences to the number of user changes. We are currently working on extending this to more complex operations such as nesting and exploding nested workflows. Future work will involve identifying semantic changes at a more conceptual level that would let us, for example, compare two versions of a workflow with a shared base and identify that two different collaborators have made similar changes.

REFERENCES

- Bolger, M. *et al.* (2016) ,Workspace: a fast and low cost methodology for delivering commercial applications based on Research IP, *eResearch Australasia*, 6–10.
- Cleary, P, Watkins, D., Hetheron, L., Bolger, M. and Thomas D. (2017). Opportunities for workflow tools to improve translation of research into impact, *22nd Int. Congr. Modelling Simulation*, 1–7.
- Oakes, N , Hetheron, L , Bolger, M , Thomas, D , Rucinski, C , Watkins, D , Cleary, P (2019). Workspace – a Scientific Workflow System with commercial impact, *23rd Int. Congr. Modelling Simulation*, [1]
- Oakes,, N, Thomas, D (2021) Workspace Workflow Comparison, *24th Int. Congr Modelling Simulation*
- Watkins, D., Thomas D., Hetheron, L., Bolger, M. and Cleary, P.W., (2017). Workspace – a Scientific Workflow System for enabling Research Impact, *22nd Int. Congr. Modelling Simulation*.
- Workspace Manual (2023), <https://research.csiro.au/static/workspace/docs>