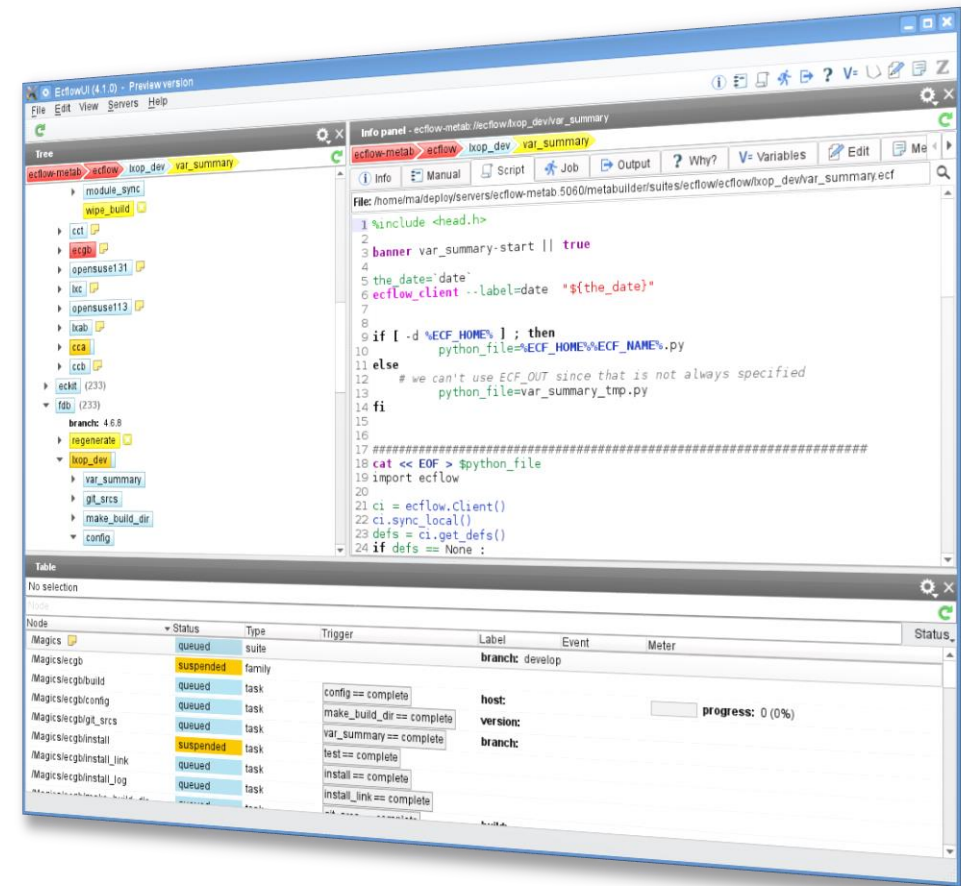


ecFlowUI - Visualising Complex Workflows

ECMWF Visualisation Week

Sándor Kertész, Iain Russell

Development Section, ECMWF



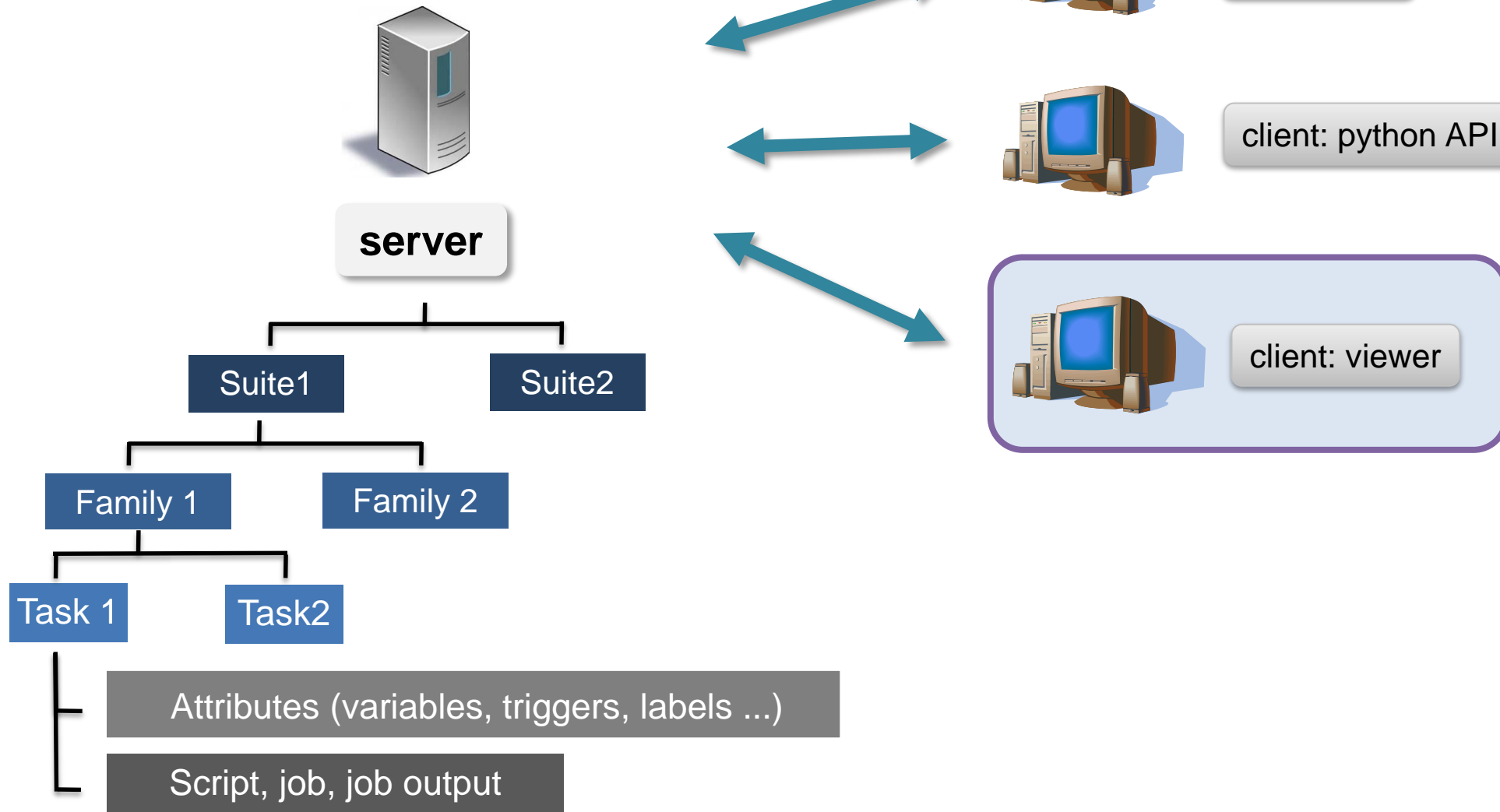
What is ecFlow?

- **Work-flow manager** developed at ECMWF
- Open source, written in C++
- Enables users to
 - design
 - run
 - monitor

a large number of tasks with **dependencies** on each other and time

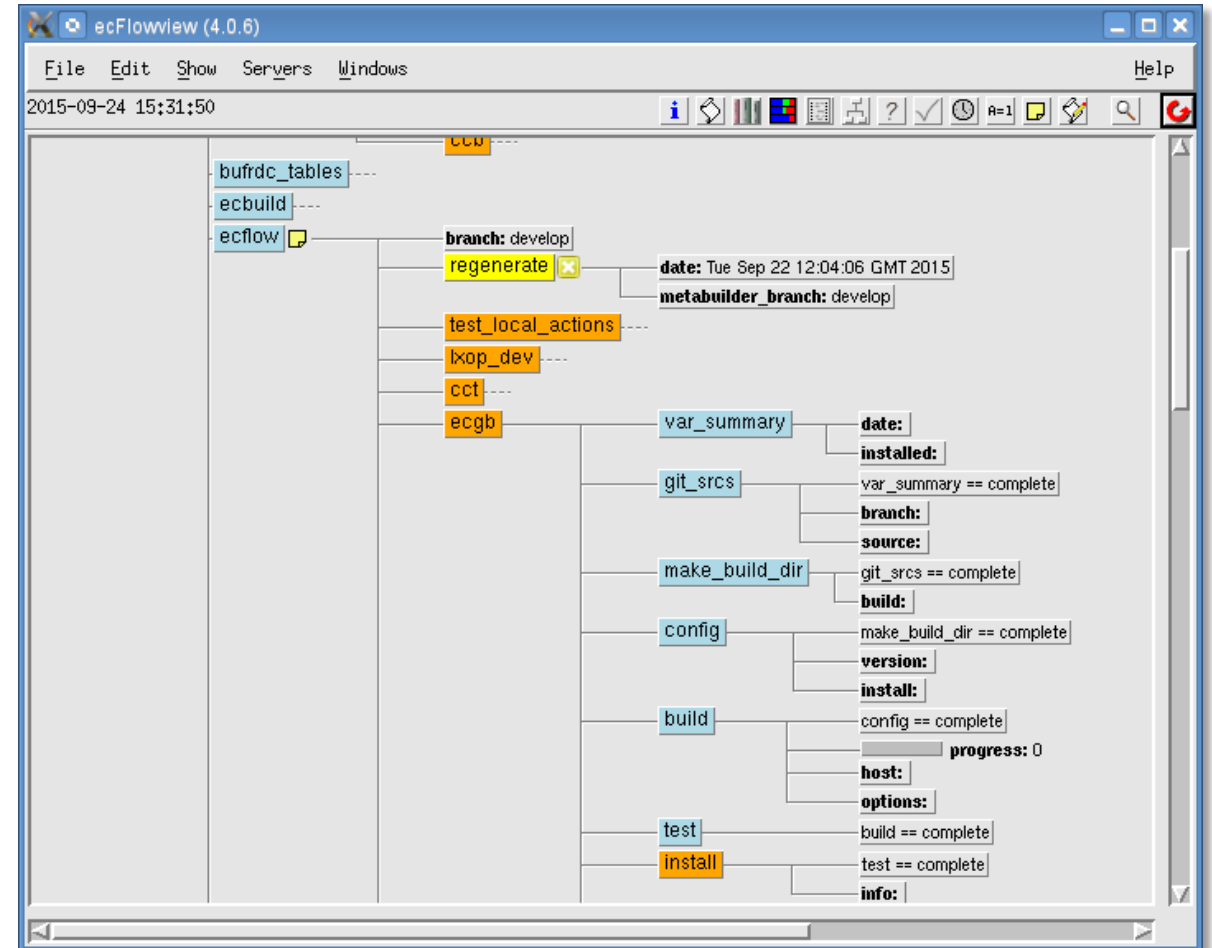
- Used for a wide range of operational, research or development activities at ECMWF other organizations. E.g.:
 - model runs
 - product generation
 - software installation

Client/Server architecture



The current viewer - ecflowview

- The primary interface to ecFlow servers
- Heavily used, highly important application
- Motif based GUI
- It was decided to develop a new viewer based on a more up to date GUI technology

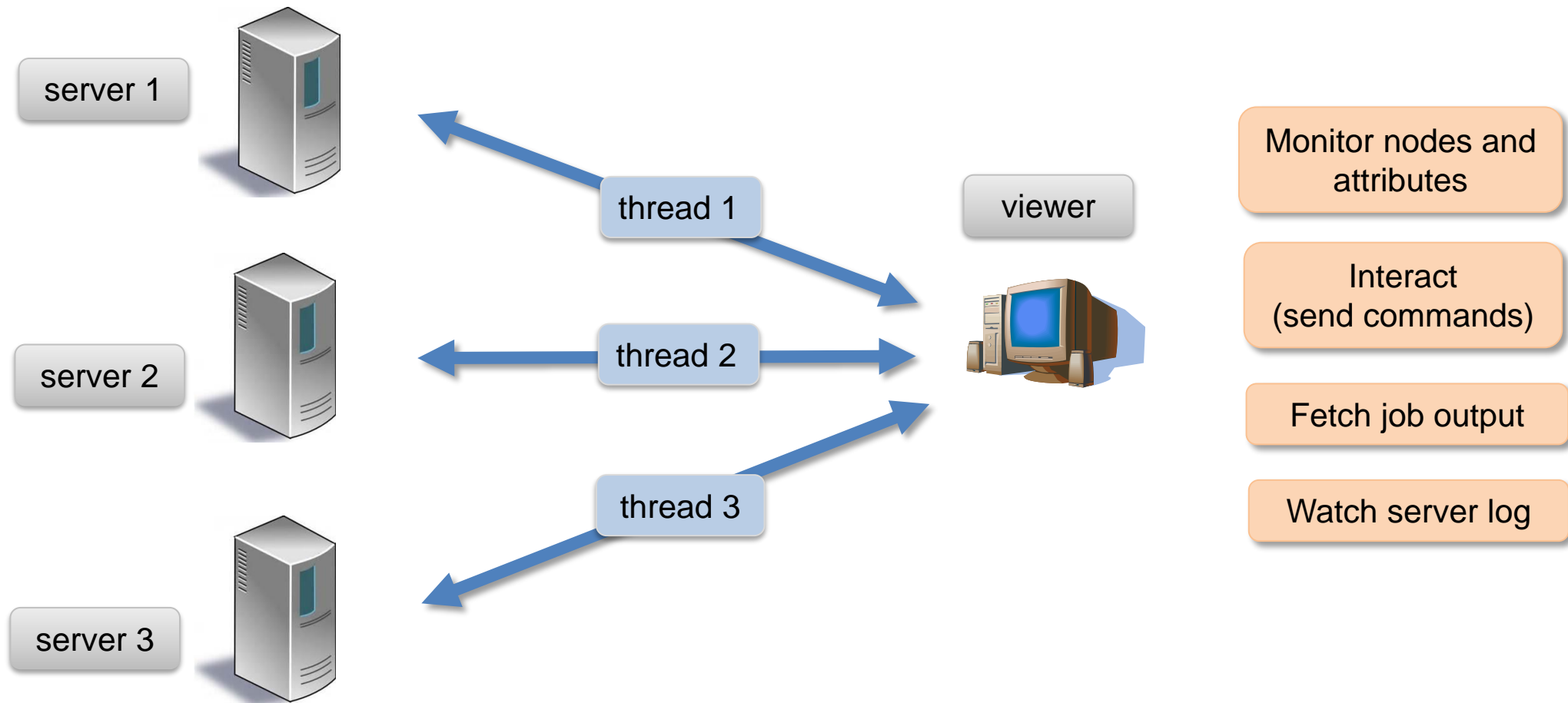


The new viewer - ecFlowUI

- Desktop application
 - Web solution was considered but desktop was favoured
- Developed with Qt and C++
 - Good experience from Metview (Motif to Qt migration)
- User requirements:
 - Various user groups had different preferences
 - Conclusion: basically all the features of the current viewer needed + need for new features
 - In terms of functionality it is very similar to the current viewer



ecFlowUI – Overview



The user interface

- Multi window, multi tabbed
- Dashboard approach

The screenshot displays the EcfloUI (4.1.0) interface, which is multi-window and multi-tabbed. The main window shows a dashboard with three panels:

- Tree View (Left):** A hierarchical tree structure of tasks and directories. The selected path is `ecflow-metab > ecflow > lxop_dev > var_summary`. Other visible nodes include `module_sync`, `wipe_build`, `cct`, `ecgb`, `opensuse131`, `lxc`, `lxab`, `cca`, `ccb`, `eckit (233)`, and `fdb (233)`. The `fdb` node is expanded to show sub-tasks like `regenerate`, `lxop_dev`, `var_summary`, `git_srcs`, `make_build_dir`, `config`, `build`, `test`, `install`, `install_log`, `remove_write_perm`, `module_sync`, `wipe_build`, `cct`, `ecgb`, and `opensuse131`.
- Table View (Center):** A table showing the status of tasks. The selected path is `ecflow-metab > libemos > opensuse113 > build`. The table has columns: Node, Status, Type, Trigger, and Label. The data is as follows:

Node	Status	Type	Trigger	Label
/mars_client/cct/gnu/make_build_dir	queued	task	git_srcs == complete	build
/mars_client/cct/gnu/tests/xtra/task00	queued	task		id:
/mars_client/cct/gnu/tests/xtra/task01	queued	task		id:
/mars_client/cct/gnu/tests/xtra/task02	queued	task		id:
/mars_client/cct/gnu/tests/xtra/task03	queued	task		id:
/mars_client/cct/gnu/tests/xtra/task04	queued	task		id:
/mars_client/cct/gnu/tests/xtra/task05	queued	task		id:
/mars_client/cct/gnu/tests/xtra/task06	queued	task		id:
- Table View (Right):** A table showing the status of tasks. The selected path is `ecflow-metab > libemos > opensuse113 > build`. The table has columns: Node, Status, Type, Trigger, and Label. The data is as follows:

Node	Status	Type	Trigger	Label
/Magics/ecgb	suspended	family		
/Magics/ecgb/install	suspended	task	test == complete	
/Magics/lxab	suspended	family		
/Magics/lxab/install	suspended	task	test == complete	
/Magics/lxc	suspended	family		
/Magics/lxc/install	suspended	task	test == complete	
/Magics/lxop_dev	suspended	family		

Tree view

- The main view to show the node structure
- In practice the tree can be very large: more than 500000 tasks each having several attributes
- Challenges:
 - Memory management
 - Update

The screenshot displays a 'Tree' window with a breadcrumb trail at the top: `ecflow-metab > ecflow > lxop_dev > var_summary`. The tree structure is as follows:

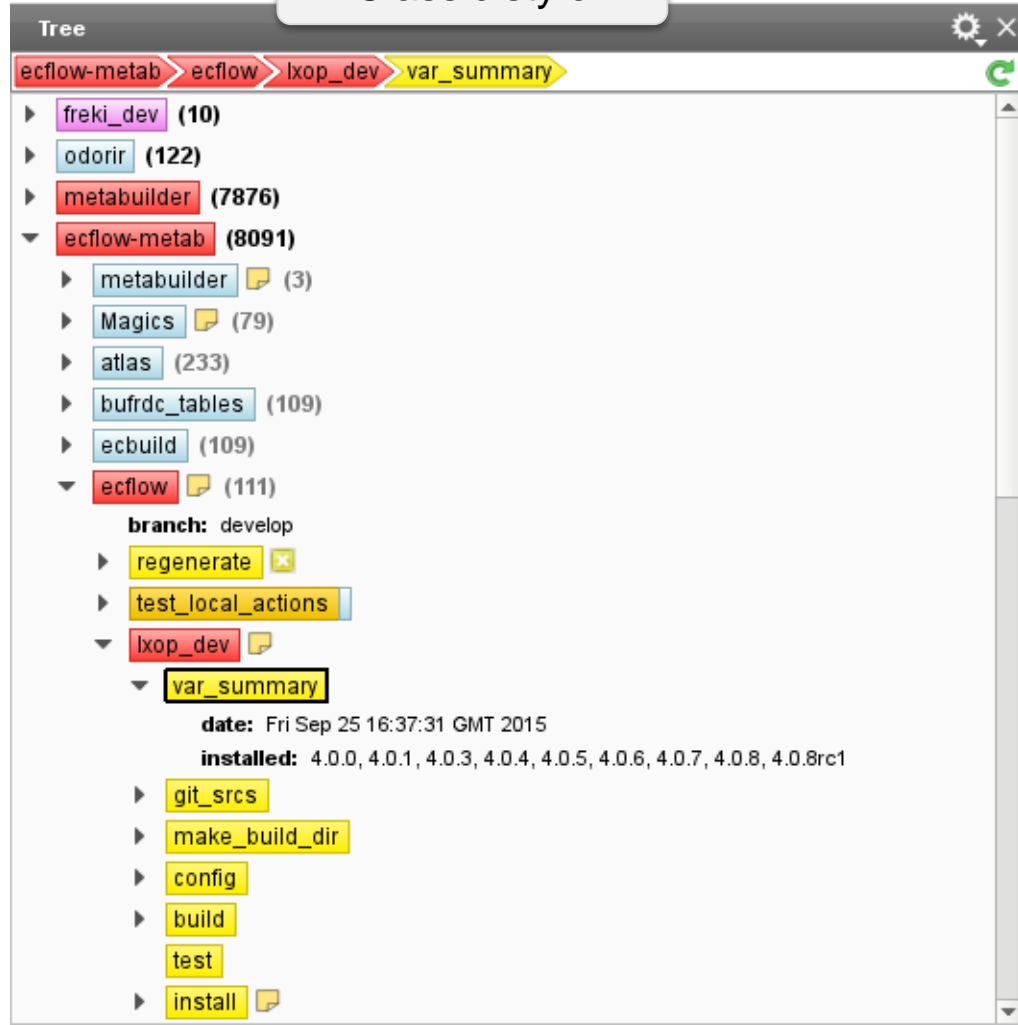
- ecflow-metab (8091)**
 - metabuilder (3)
 - Magics (79)
 - atlas (233)
 - bufrdc_tables (109)
 - ecbuild (109)
 - ecflow (111)**
 - branch: develop
 - regenerate
 - test_local_actions
 - lxop_dev**
 - var_summary**
 - date: Fri Sep 25 10:16:38 GMT 2015
 - installed: 4.0.0, 4.0.1, 4.0.3, 4.0.4, 4.0.5, 4.0.6, 4.0.7, 4.0.8, 4.0.8rc1
 - git_srcs
 - make_build_dir
 - config
 - build**
 - host: lxop-dev
 - options: -j16
 - progress: 100 (100%)
 - config == complete
 - test
 - install

Annotations in the image point to various elements:

- breadcrumbs**: Points to the breadcrumb trail at the top.
- server**: Points to the `ecflow-metab` node.
- suite, family and task nodes**: Points to the `regenerate`, `test_local_actions`, and `lxop_dev` nodes.
- node attributes**: Points to the `host`, `options`, and `progress` attributes of the `build` node.

Tree view – two styles

Classic style

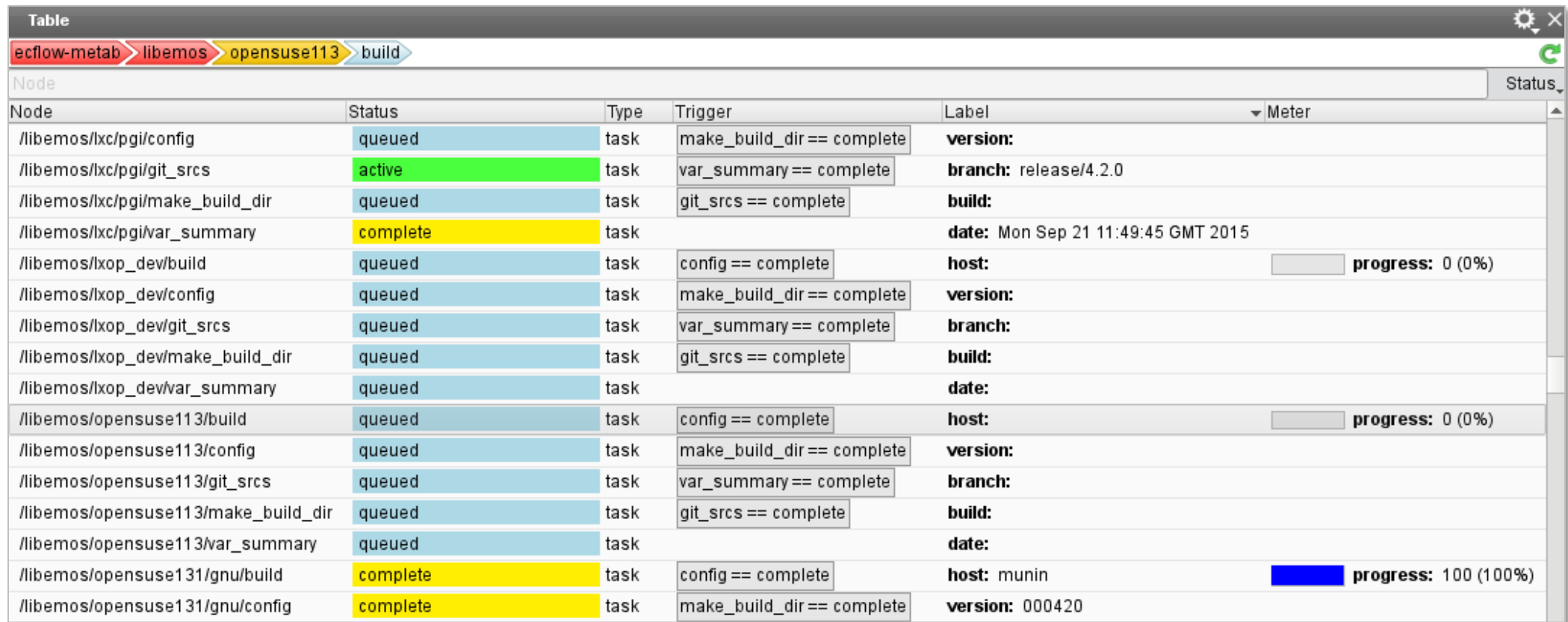


New style



Table view

- More suitable for sorting/filtering than the tree
- Configurable columns, custom filters
- Challenging to update (filter has to be run all the time)

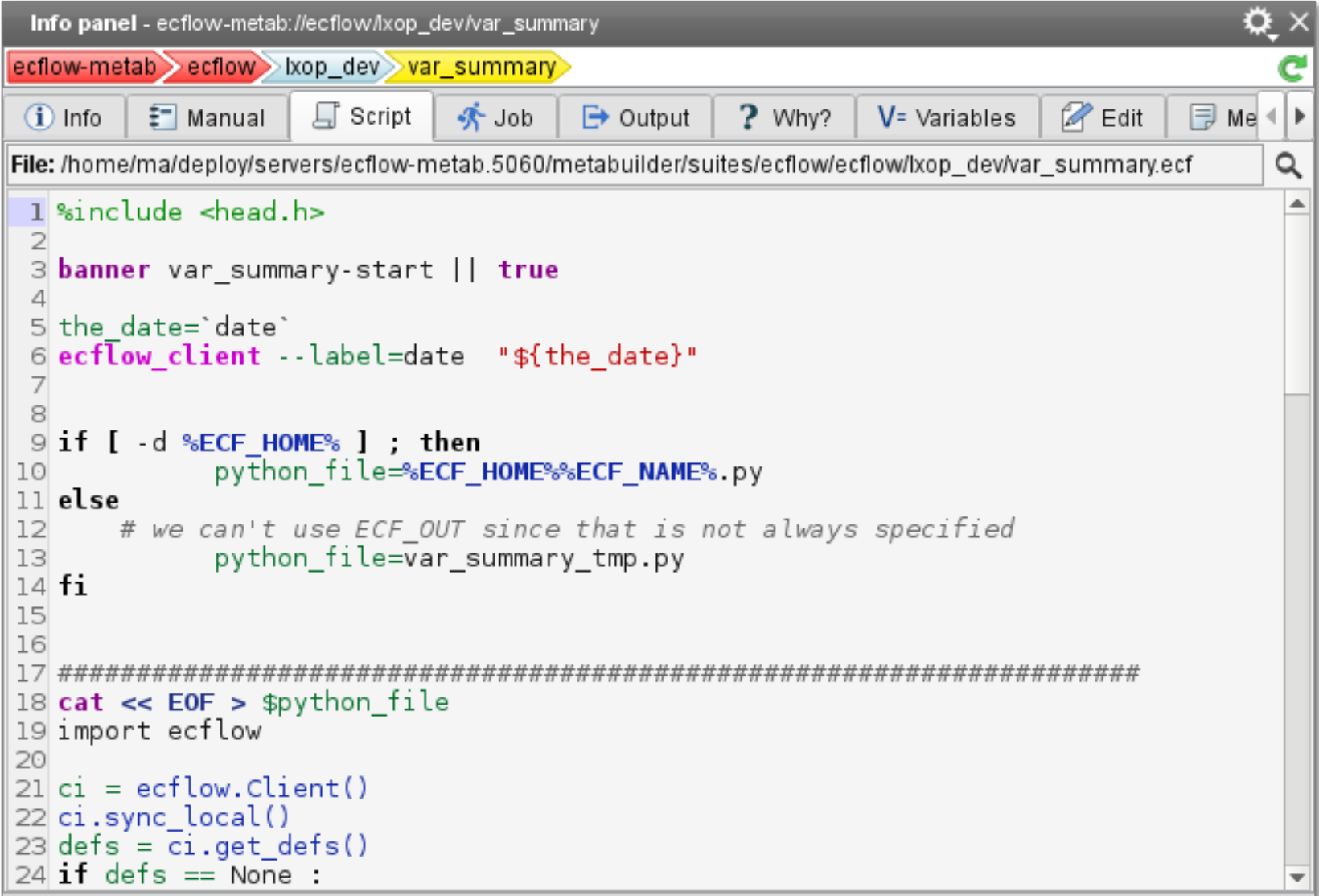


The screenshot shows a web interface for a task table. At the top, there are breadcrumb navigation links: 'ecflow-metab' (red), 'libemos' (orange), 'opensuse113' (yellow), and 'build' (blue). Below the breadcrumbs is a 'Node' search bar and a 'Status' dropdown. The main table has columns for 'Node', 'Status', 'Type', 'Trigger', 'Label', and 'Meter'. The 'Status' column uses color coding: blue for 'queued', green for 'active', and yellow for 'complete'. The 'Label' column contains various task-specific information like 'version:', 'branch:', 'build:', and 'date:'. The 'Meter' column shows progress bars and percentages for tasks that are 'queued' or 'complete'.

Node	Status	Type	Trigger	Label	Meter
/libemos/lxc/pgi/config	queued	task	make_build_dir == complete	version:	
/libemos/lxc/pgi/git_srcs	active	task	var_summary == complete	branch: release/4.2.0	
/libemos/lxc/pgi/make_build_dir	queued	task	git_srcs == complete	build:	
/libemos/lxc/pgi/var_summary	complete	task		date: Mon Sep 21 11:49:45 GMT 2015	
/libemos/lxop_dev/build	queued	task	config == complete	host:	progress: 0 (0%)
/libemos/lxop_dev/config	queued	task	make_build_dir == complete	version:	
/libemos/lxop_dev/git_srcs	queued	task	var_summary == complete	branch:	
/libemos/lxop_dev/make_build_dir	queued	task	git_srcs == complete	build:	
/libemos/lxop_dev/var_summary	queued	task		date:	
/libemos/opensuse113/build	queued	task	config == complete	host:	progress: 0 (0%)
/libemos/opensuse113/config	queued	task	make_build_dir == complete	version:	
/libemos/opensuse113/git_srcs	queued	task	var_summary == complete	branch:	
/libemos/opensuse113/make_build_dir	queued	task	git_srcs == complete	build:	
/libemos/opensuse113/var_summary	queued	task		date:	
/libemos/opensuse131/gnu/build	complete	task	config == complete	host: munin	progress: 100 (100%)
/libemos/opensuse131/gnu/config	complete	task	make_build_dir == complete	version: 000420	

Info panel

- Node specific
- Displays node attributes, logs, outputs



The screenshot shows a web-based interface for an ECFLOW node. The title bar reads "Info panel - ecflow-metab://ecflow/lxop_dev/var_summary". Below the title bar is a breadcrumb trail: "ecflow-metab > ecflow > lxop_dev > var_summary". A toolbar contains icons for "Info", "Manual", "Script", "Job", "Output", "Why?", "Variables", "Edit", and "Me". The main content area displays the file path: "/home/ma/deploy/servers/ecflow-metab.5060/metabuilder/suites/ecflow/ecflow/lxop_dev/var_summary.ecf". The file content is as follows:

```
1 %include <head.h>
2
3 banner var_summary-start || true
4
5 the_date=`date`
6 ecflow_client --label=date "${the_date}"
7
8
9 if [ -d %ECF_HOME% ] ; then
10     python_file=%ECF_HOME%%ECF_NAME%.py
11 else
12     # we can't use ECF_OUT since that is not always specified
13     python_file=var_summary_tmp.py
14 fi
15
16
17 #####
18 cat << EOF > $python_file
19 import ecflow
20
21 ci = ecflow.Client()
22 ci.sync_local()
23 defs = ci.get_defs()
24 if defs == None :
```

Job output

- Job outputs can be huge (>1 GB). It is challenging to display such big text files in a Qt text editor.

```
1444203 *** SCRIPT TIMINGS ****
1444204 Total time : 960.65 seconds
1444205
1444206 Script time : 960.65 seconds
1444207 Parallel run_parallel time : 0.00 seconds
1444208 Serial run_parallel_time : 0.00 seconds
1444209
1444210 Parallel run_parallel count : 0
1444211 Serial run_parallel count : 0
1444212
1444213 Script percentage : 100%
1444214 Parallel run_parallel percentage : 0%
1444215 Serial run_parallel percentage : 0%
1444216 ****
1444217
1444218 ****
1444219 *** Individual run_parallel data ****
1444220 Started Pxt Seconds Command
1444221 cat: /scl/TMPDIR/emos/JTMP/06/emos.6901680.ccbpp2.ccbppn041.201509
1444222 ****
1444223
1444224
1444225 ****
1444226 *** slow lines of script ****
1444227 ****
```

Name	Size	Modified	Modified
odbtools.1	128 MB	21 hours ago	2015-09-24 16:25:19
odbtools.job1.sub	15 B	21 hours ago	2015-09-24 16:09:11
odbtools.job1	161 KB	21 hours ago	2015-09-24 16:09:10

Plans

- ecFlowUI is still in development
- There are still features to add e.g.:
 - Search within the node hierarchy
 - Dependency graph
- **Alpha** version is already available internally
 - will be available externally in ecFlow 4.1.0
- Planned to make it feature complete in 2016

Questions?