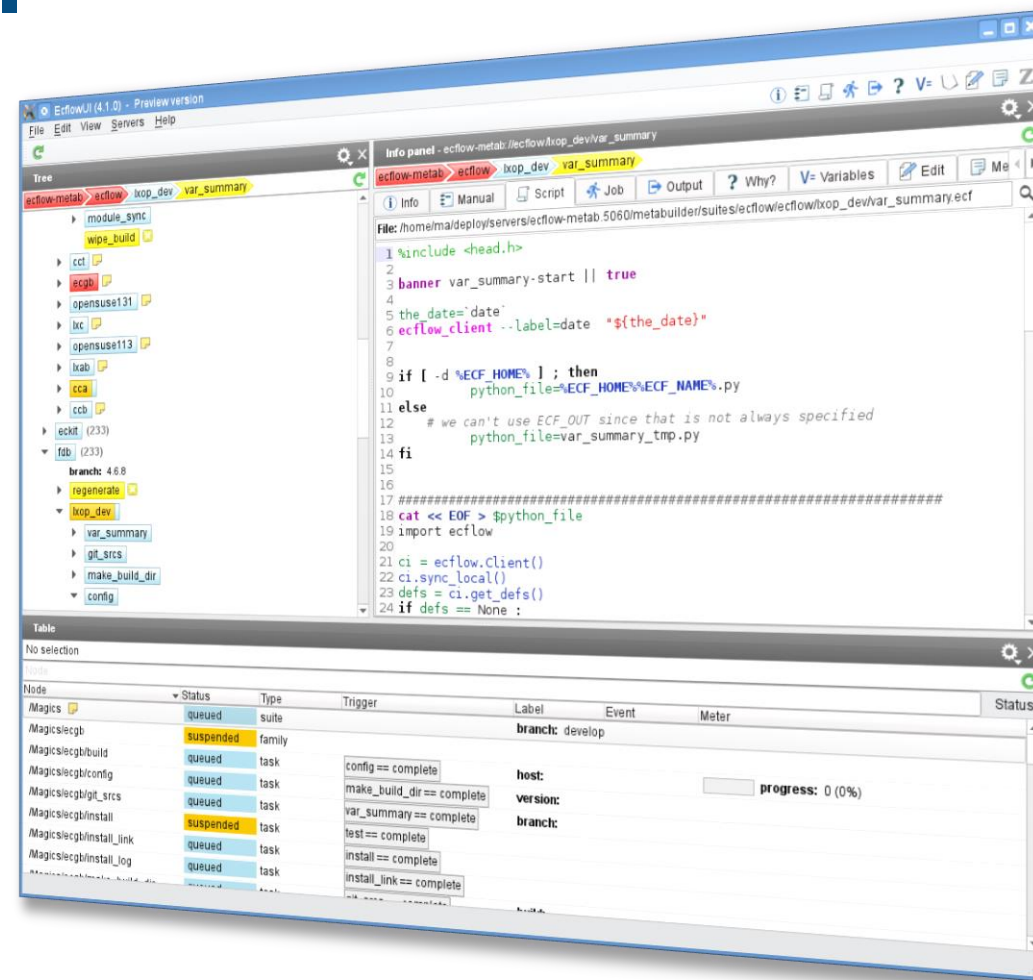


Update on ecFlowUI

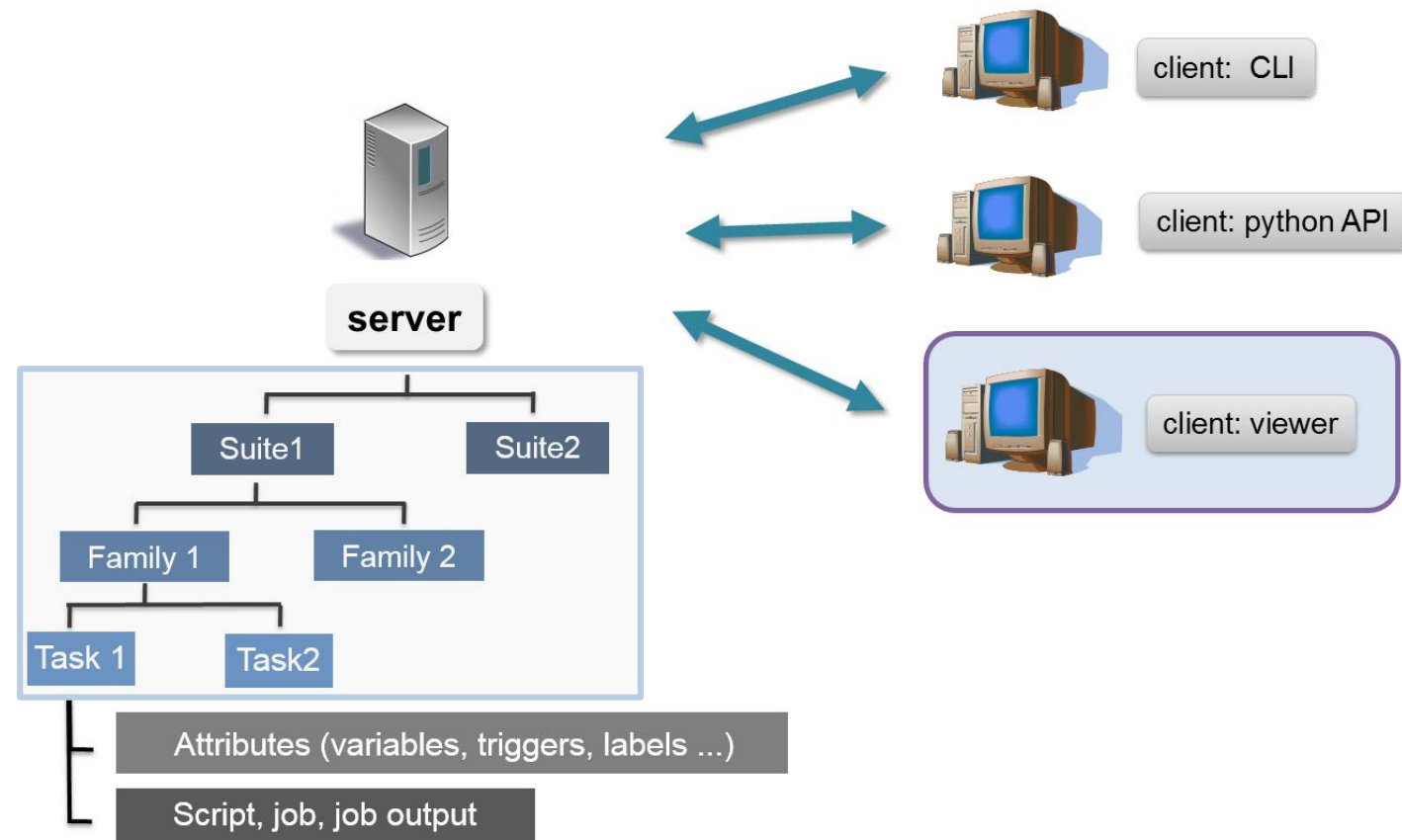
Sándor Kertész, Iain Russell

Development Section, ECMWF



What is ecFlow?

- **Work-flow manager** developed at ECMWF
- Enables users to
 - design
 - run
 - monitora large number of tasks with **dependencies** on each other and time
- Heavily used at ECMWF in
 - operations
 - research
 - development
- Used at other organisations as well



ecFlowUI

- New graphical user interface for ecFlow
- To monitor tasks and interact with them
- Dashboard approach

Tree view

Table view

The screenshot displays the ecFlowUI interface with a menu bar (File, Edit, View, Refresh, Servers, Tools, Help) and a toolbar. The main window is divided into three panes:

- Tree view:** A hierarchical tree of tasks. The selected path is `local-mega > operation_suite > 0 > analysis > run_analysis`. The tree shows nodes like `metabuilder`, `local-mega`, `metview`, `operation_suite`, `day=1`, `F 0`, `analysis`, `post_processing`, `get_observations`, `run_analysis` (with sub-nodes `info: get obs`, `info2: get obs2`, `get_observations == complete`), `forecast`, `archive`, `12`, and `why_test`.
- Info panel:** A panel titled `Info panel - local-mega://operation_suite/0/analysis/run_analysis`. It contains a code editor with the following content:

```
10 export ECF_NAME=/operation_suite/0/analysis/run_analysis # The name of this current task
11 export ECF_PASS=VUg9qD.0 # A unique password
12 export ECF_TRYNO=9 # Current try number of the task
13 export ECF_RID=$$
14
15 echo "Running on $HOST"
16
17 # Define the path where to find ecflow_client
18 # make sure client and server use the *same* version.
19 # Important when there are multiple versions of ecFlow
20 export PATH=/usr/local/apps/ecflow/4.3.0/bin:$PATH
21
22 # Tell ecFlow we have started
23 ecflow_client --init=$$
24
25
26 # Define a error handler
27 ERROR() {
28     set +e # Clear -e flag, so we don't fail
29     ecflow_client --abort=trap # Notify ecFlow that something went wrong, using 'trap' as t
30     trap 0 # Remove the trap
31     exit 0 # End the script
32 }
```
- Table view:** A table titled `Table` with a filter `SELECT node WHERE (active)`. The table has columns: Node, Status, Type, Trigger, Label, Event, and Meter. The data is as follows:

Node	Status	Type	Trigger	Label	Event	Meter
/operation_suite/12/archive/analysis/step_12/save	active	task				
/operation_suite/12/archive/analysis/step_12	active	family		../forecast/run_forecast:step ge 12		
/operation_suite/12/archive/analysis	active	family		./analysis/run_analysis == complete		
/operation_suite/12/archive	active	family				
/operation_suite/12	active	family		/0 == complete		
/operation_suite	active	suite				

Search interface

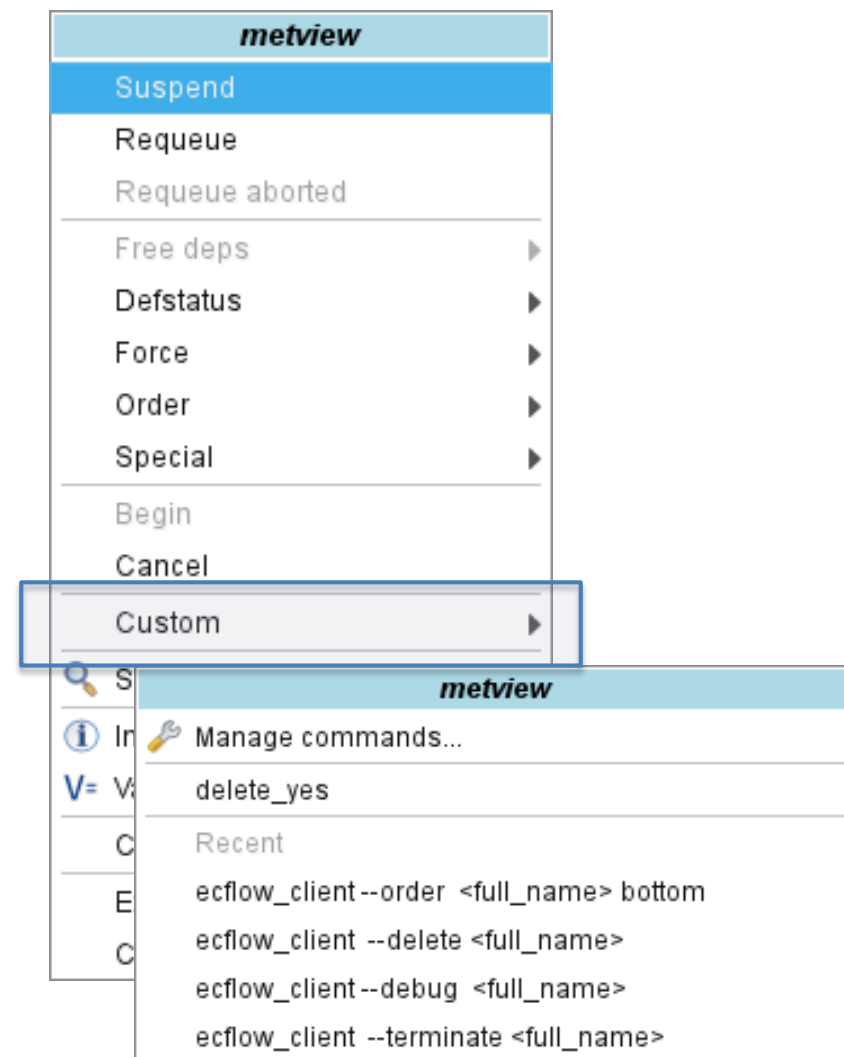
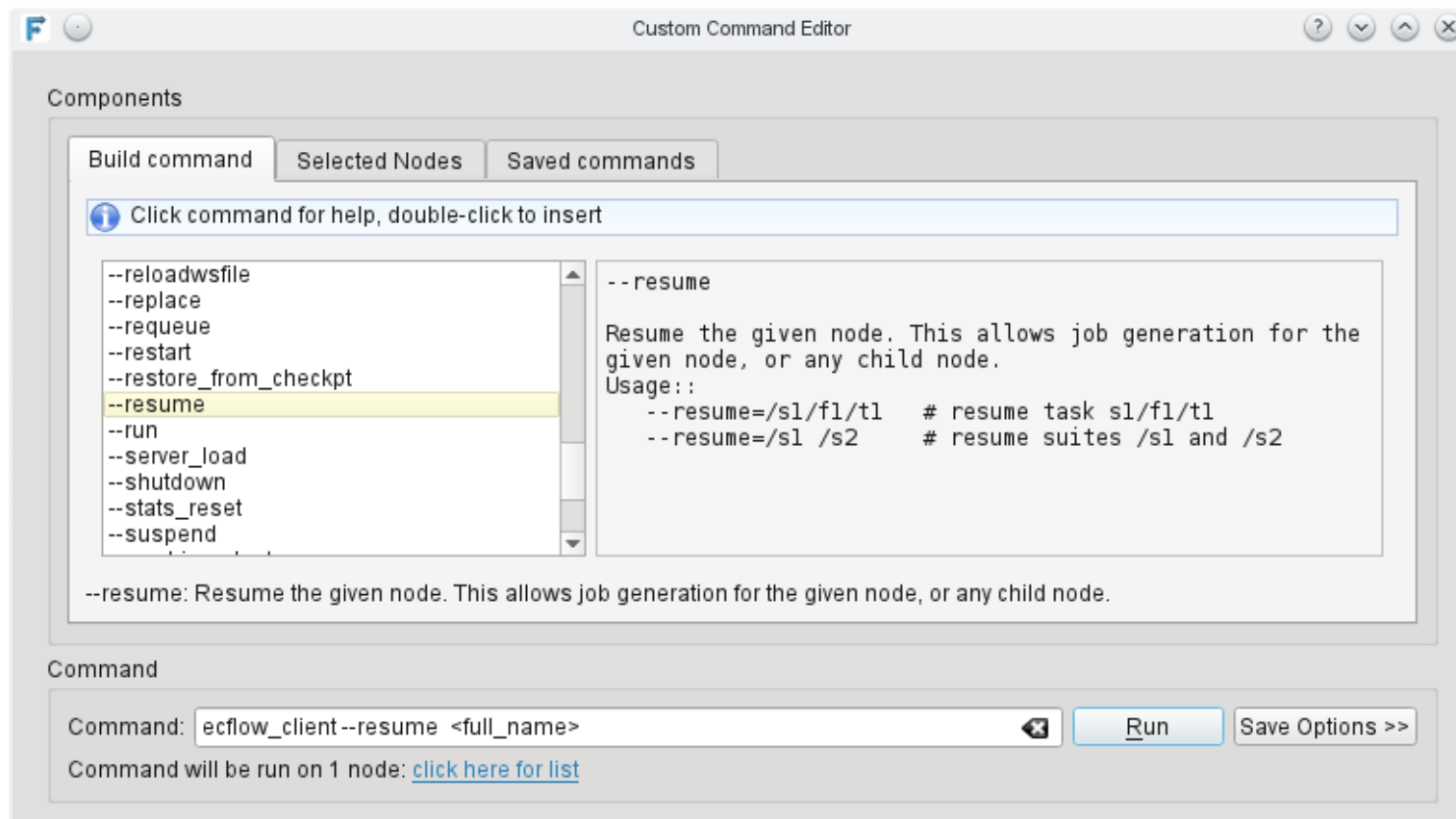
- Powerful interface to find nodes and attributes
- Can run commands straight on the results

The screenshot displays the ECFLOW search interface. At the top, the 'Scope' section shows 'Search in servers: ecflow-metab' and 'Search root node:'. The 'Global options' section includes 'Max results: 50000' and a 'Case sensitive' checkbox. A search bar with a magnifying glass icon and a 'Search' button is present. Below this, there are tabs for 'Nodes*' and 'Attributes', with a 'Show:' button and an 'Editor' button. The 'Name:' field is set to 'Contains' with the value 'ANY'. The 'Path:' field is set to 'Matches' with the value '*emos*'. There are three columns of filters: 'Type', 'Status', and 'Flag'. The 'Type' column has checkboxes for 'server', 'suite', 'family', 'task', and 'alias', with 'suite', 'family', and 'task' checked. The 'Status' column has checkboxes for 'aborted', 'active', 'complete', 'queued', 'submitted', 'suspended', and 'unknown', with 'aborted', 'complete', and 'queued' checked. The 'Flag' column has checkboxes for 'is_late', 'has_da', 'has_me', 'has_tim', 'is_rerun', 'is_waiti', and 'is_zoml'. A 'Query' section shows the SQL query: `SELECT node FROM ecflow-metab WHERE (node_path = '*emos*') and (suite or family or task) and (aborted or complete or queued) LIMIT 50000`. Below the query is a table with columns 'Server', 'Node', 'State', and 'Type'. The table contains 10 rows of search results. At the bottom, it says '9392 items found in 0.1 s' and has a 'Close' button.

Server	Node	State	Type
ecflow-metab	/libemos/opensuse131/gnu.48/module	complete	task
ecflow-metab	/libemos/opensuse131/gnu.48/remove	complete	task
ecflow-metab	/libemos/opensuse131/gnu.48/test	aborted	task
ecflow-metab	/libemos/opensuse131/gnu.48/var_sun	complete	task
ecflow-metab	/libemos/opensuse131/gnu.48/wipe_bu	queued	task
ecflow-metab	/libemos/opensuse131/gnu.53	aborted	family
ecflow-metab	/libemos/opensuse131/gnu.53/add_wri	complete	task
ecflow-metab	/libemos/opensuse131/gnu.53/build	complete	task
ecflow-metab	/libemos/opensuse131/gnu.53/config	complete	task
ecflow-metab	/libemos/opensuse131/gnu.53/git_srcs	complete	task

Custom command editor

- Node context menu provides node-sensitive commands
- Need for an editor to create and run commands



Notifications

Configuration

The screenshot shows the 'Notifications' configuration window. On the left is a sidebar with icons for Appearance, Fonts, Notifications (highlighted), and Server options. The main area is titled 'Notifications' and contains the following settings:

- Maximum notification list size: 200 items
- Notification types: Aborted, Restarted, Late, Zombies, Aliases (selected)
- Description: Notify when a task becomes aborted
- Main settings:
 - Enabled:
 - Popup dialog:
 - Play sound:
 - Note: These settings can be customised for each server separately
- Colour:
 - Notification count background colour: [blue swatch]
 - Notification count text colour: [white swatch]
 - Note: For aborted notifications the background and text colours are taken from the Status colour settings
- Sound:
 - Sound file type: Built in
 - Sound loop count: 1

Buttons at the bottom: Apply, Cancel, OK.

The screenshot shows the main application interface with a 'Change notification' popup window and a notification bar at the bottom.

The 'Change notification' window has tabs for Aborted (5), Aliases, Late, Restarted, and Zombies. It contains a table with the following data:

Server	Node	Time of change
freki_dev	/suite_2/family_1/task_a	2017-02-27 11:38:44
freki_dev	/suite_2/family_1/task_c	2017-02-27 11:38:46
freki_dev	/suite_2/family_1/task_b	2017-02-27 11:38:46
freki_dev	/suite_2/family_1/task_d	2017-02-27 11:38:46
freki_dev	/suite_2/family_1/task_e	2017-02-27 11:38:46

Below the table are checkboxes for 'Clear current list on close' and buttons for 'Clear' and 'Close'. A blue arrow points from the 'Popup dialogue' label to the 'Clear' button.

The main application window shows a tree view with a selected node 'suite_2' (6) containing tasks task_1, task_2, and a sub-tree 'family_1' with tasks task_a, task_c, task_b, task_d, and task_e. A blue arrow points from the 'Notification bar' label to the notification bar at the bottom.

The notification bar at the bottom contains icons for Aborted (A5), Aliases (As), Late (L), Restarted (Re), and Zombies (Z).

Popup dialogue

Notification bar

Trigger view

- Triggers expressions define when tasks can be run

Dependencies

Triggers of the selected node Nodes triggered by the selected node

Trigger expression

```
(/e_43r1/main:YMD gt /e_43r1/lag:YMD) or (/e_43r1/main:YMD eq /e_43r1/lag:YMD and /e_43r1/main/12/an==complete and /e_43r1/main/12/fc==complete) AND (/e_43r1/main:YMD gt /e_43r1/lag:YMD) or (/e_43r1/main:YMD eq /e_43r1/lag:YMD and /e_43r1/main/12/an eq complete and (/e_43r1/main/12/fc/model:step ge 9 or /e_43r1/main/12/fc/model eq complete))
```

Triggers directly triggering the selected node

genvar	/e_43r1/main:YMD
genvar	/e_43r1/lag:YMD
family	/e_43r1/main/12/an
family	/e_43r1/main/12/fc
family	/e_43r1/main/12/an
meter	/e_43r1/main/12/fc/model:step
task	/e_43r1/main/12/fc/model

Triggers through parent suite /e_43r1

limit	/e_43r1/limits:hpc
limit	/limits:hpc

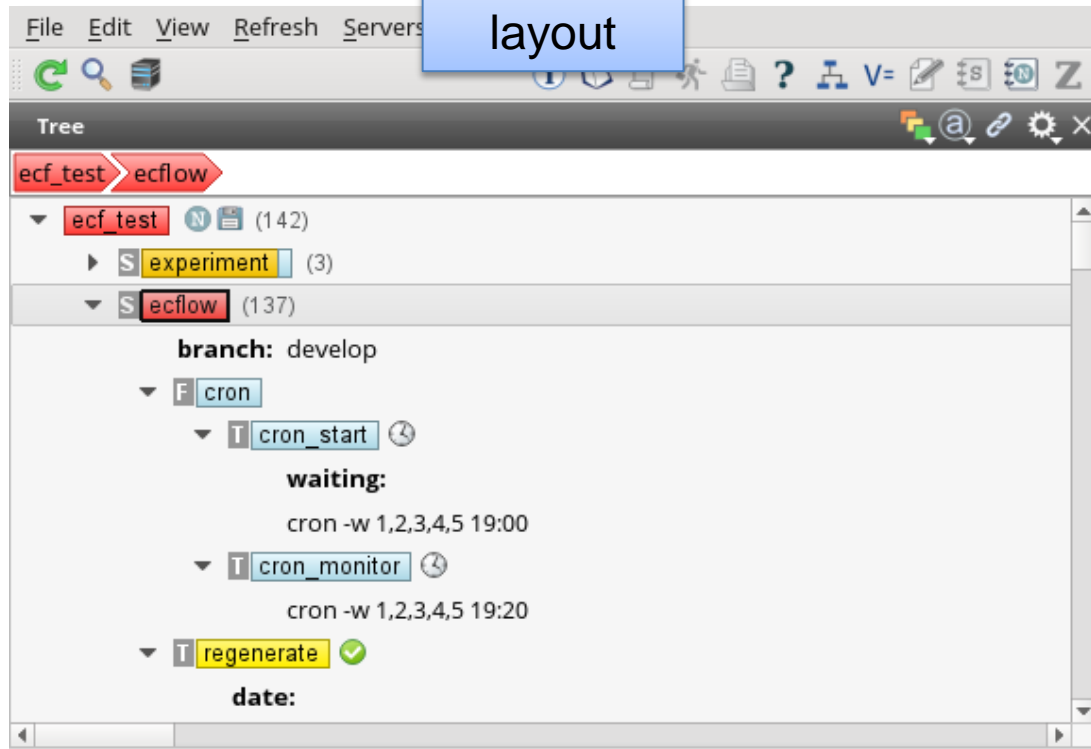
Triggers through child task /e_43r1/lag/12/archive/ansfc

limit	/e_43r1/limits:mars
limit	/limits:mars

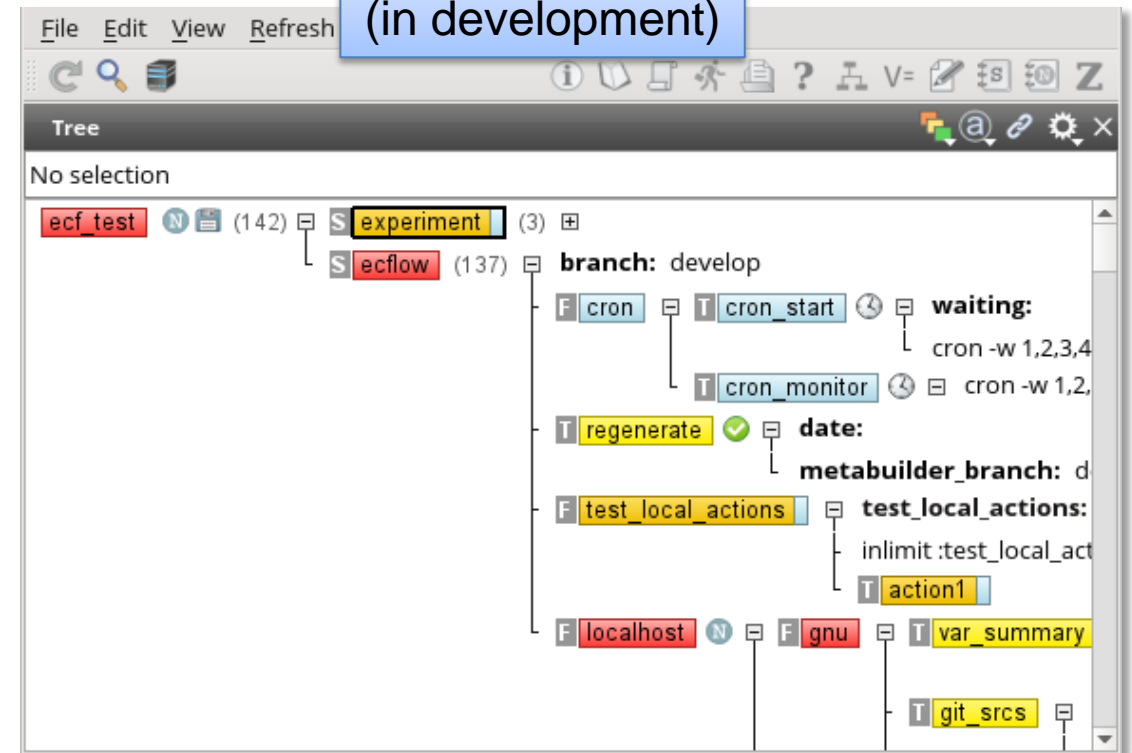
Compact tree layout (in development)

- The standard tree view mostly uses the vertical space
- Compact tree layout: better usage of horizontal space – many users prefer it

Standard layout



Compact layout (in development)



Current status

- ecFlowUI is in development
- Externally released as part of ecFlow since 2016 September (current version is 4.5.0)
- Used daily by many at ECMWF
- Remaining major requirements for full operational usage:
 - Improved trigger view
 - Compact tree layout
- Plan to make it feature complete this year

Questions?