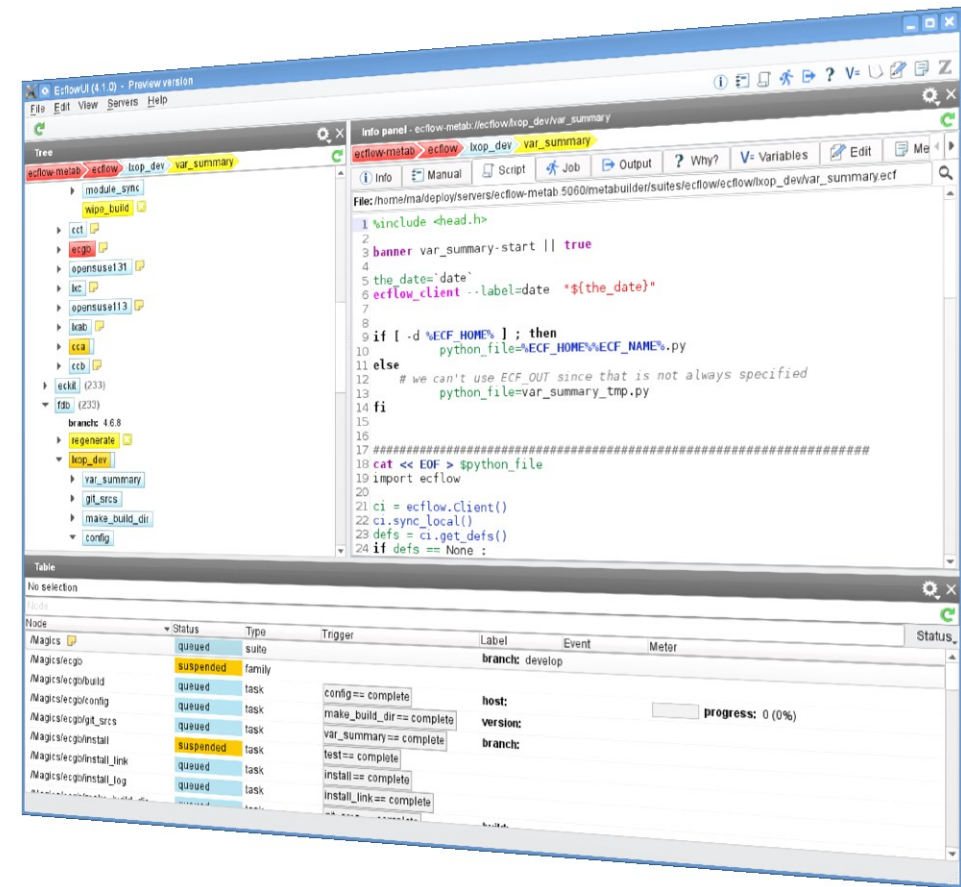


ecFlowUI

ecFlow training course 2017

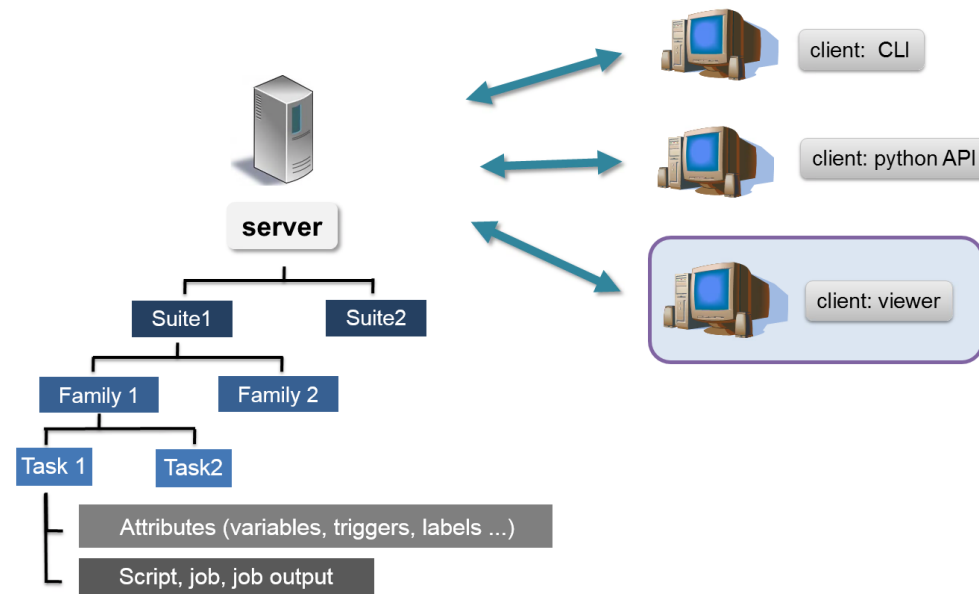
Iain Russell, Sándor Kertész

Development Section, ECMWF



What is ecFlowUI?

- Graphical user interface to **ecFlow**
- Developed at ECMWF
- Displays and allows interaction with ecFlow suites



The screenshot displays the ecFlowUI interface. The top part shows a tree view of the job structure, including nodes like 'metabuilder', 'local-mega', 'metview', 'operation_suite', 'analysis', 'post_processing', 'get_observations', 'run_analysis', 'forecast', 'archive', and 'why_test'. The 'run_analysis' task is selected, and its script is shown in the code editor. The script includes environment variables and a call to 'ecflow_client'.

```
10 export ECF_NAME=/operation_suite/0/analysis/run_analysis # The name of this current task
11 export ECF_PASS=Vlg9qD.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 }
```

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_forecaststep 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				

Transition from ecflowview

- **ecflowview** is the original user interface for ecFlow
- **ecFlowUI** is becoming its replacement
- Should be easier to use
- Should be more responsive
- Should have lower memory usage

The image displays two overlapping windows from the ecFlow ecosystem. The top window is 'ecFlowview (4.0.6)', showing a hierarchical tree of tasks and their dependencies. The bottom window is 'ecFlowUI (4.5.0) - Preview version', which provides a more detailed view of a specific task's configuration and execution results.

The 'ecFlowUI' window shows a tree view on the left with the following structure:

- mdev (795)
 - eccodes (627)
 - sappa (159)
 - var_summary (157)

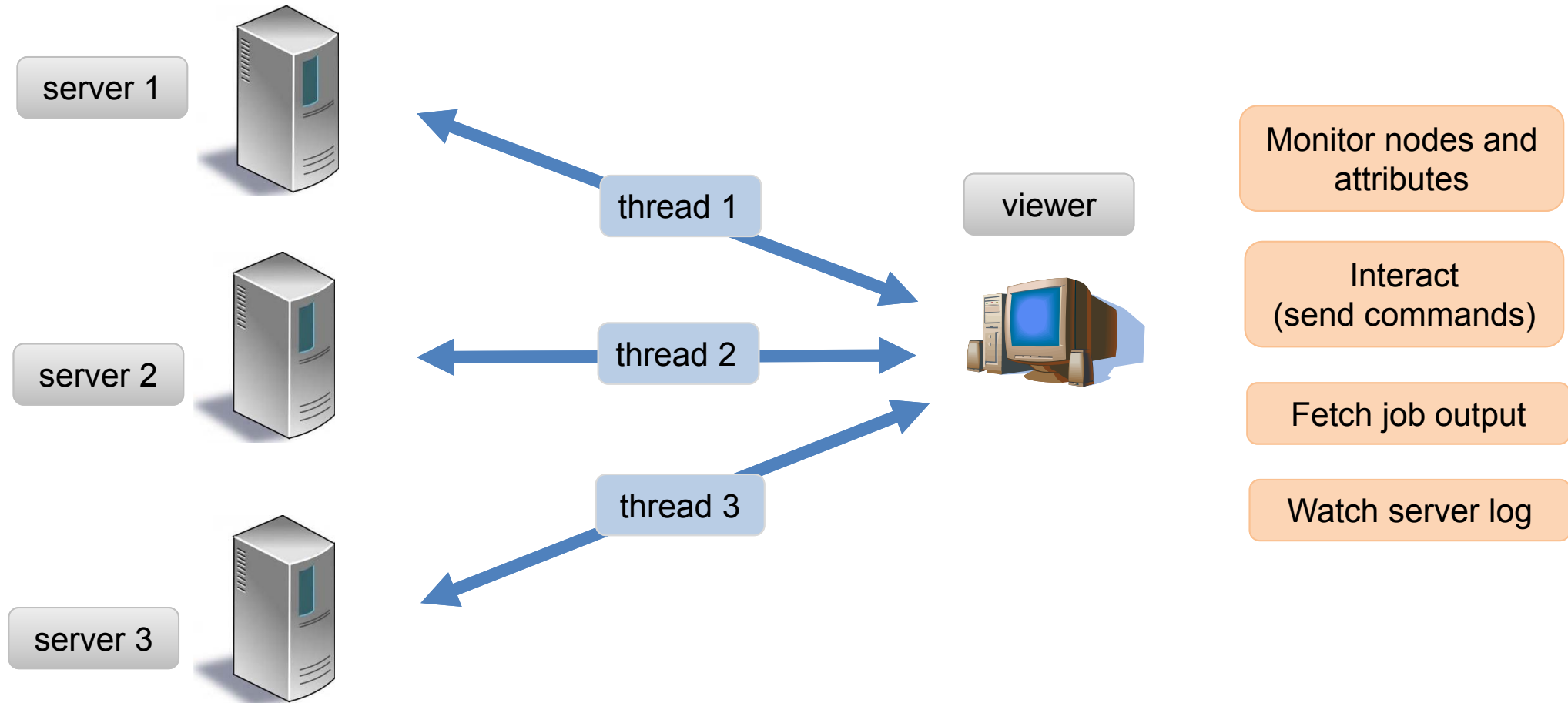
The 'Info panel' on the right shows the configuration for the 'var_summary' task, including the file path and the source code:

```
File: /home/ma/deploy/servers/ecflow-metab.5060/metabuilder/suites/eccodes/eccodes/sappa/var_summary.ecf
Source: fetched from server at 2017-01-17 13:47:18
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 :
25     print "No defs !!!!!"
26     exit(1)
27
28 node = defs.find_abs_node("%ECF_NAME%")
29 if node == None:
30     print "Could not find task %ECF_NAME%"
```

The 'Table' at the bottom shows a list of task execution results:

Node	Status	Type	Trigger	Label	Event	Meter	Status changed
/eccodes/leap42/gnu.53/	aborted	task	install_check == complete				2017-Jan-05 17:19:00
/eccodes/leap42/gnu.61	aborted	family					2017-Jan-05 17:19:00
/eccodes/leap42/gnu.61/	aborted	task	install_check == complete				2017-Jan-05 17:19:00
/eccodes/leap42/pgi	aborted	family					2017-Jan-05 17:19:00
/eccodes/leap42/pgi/inst	aborted	task	install_check == complete				2017-Jan-05 17:19:00
/atlas_bundle	aborted	suite		branch: 0.8.2.0			2016-Dec-09 16:37:00
/atlas_bundle/kg	aborted	family					2016-Dec-09 16:37:00
/atlas_bundle/kg/gnu.53	aborted	family					2016-Dec-09 16:37:00

ecFlowUI – Overview



The user interface

- Multi window, multi tabbed
- Dashboard approach
- Can add any number of panels to a dashboard
- Panel can be a Tree View, a Table View or an Info Panel
- Each dashboard communicates with its own set of servers
- Tab headers show server status

The screenshot displays the ECFLOW user interface, which is a multi-window, multi-tabbed dashboard. The interface is organized into several panels:

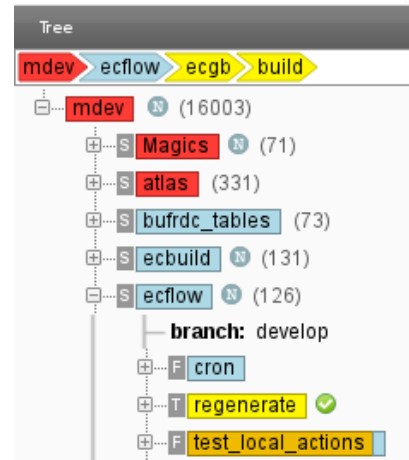
- Tree View:** A hierarchical tree structure on the left side, showing the workflow. The selected path is `local-mega > operation_suite > 0 > analysis > run_analysis`. The tree includes nodes like `metabuilder`, `local-mega`, `metview`, `operation_suite`, `day=1`, `0`, `analysis`, `post_processing`, `get_observations`, `run_analysis`, `forecast`, `archive`, and `why_test`.
- Info Panel:** A panel on the right showing the details of the selected task. It includes the file path `/home/graphics/cgi/ecflow/course/operation_suite/0/analysis/run_analysis.job9` and the source information. The main content is a shell script snippet:

```
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 at the bottom showing the status of various nodes. The filter is set to `SELECT node WHERE (active)`. The table has columns for Node, Status, Type, Trigger, Label, Event, and Meter.

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				

Tree view

- The main view to show the node structure
- In practice the tree can be very large: more than 500,000 tasks each having several attributes
- The 'old' tree rendering style is still available

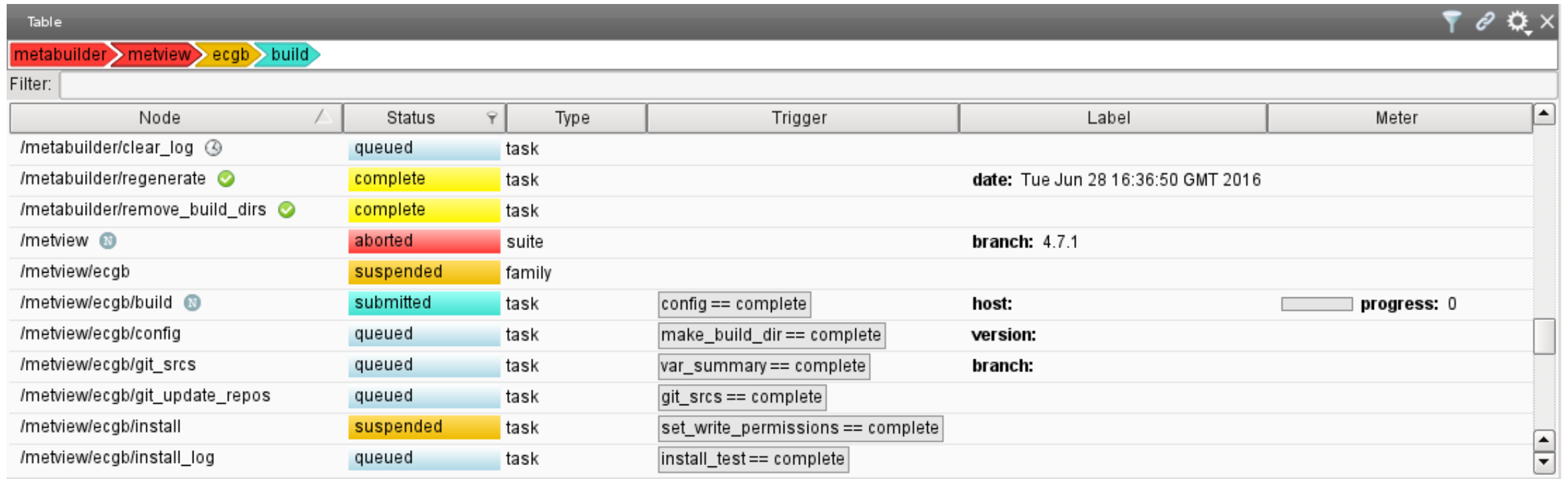


A large screenshot of the tree view interface with several blue callout boxes pointing to specific features:

- breadcrumbs**: Points to the breadcrumb path at the top: mdev > ecflow > ecgb > build.
- server**: Points to the 'mdev' node at the top of the tree.
- suite, family and task nodes**: Points to various nodes in the tree, including 'cron', 'regenerate', 'test_local_actions', and 'ecgb'.
- node attributes**: Points to the detailed view of the 'build' task node, showing attributes like 'host: ecgb04', 'options: -j32', 'progress: 100' (with a blue progress bar), and 'config == complete'.

Table view

- More suitable for sorting/filtering than the tree
- Configurable columns, powerful filtering capabilities
- Current implementation can be slow when viewing very large servers



Table

metabuilder > metview > ecgb > build

Filter:

Node	Status	Type	Trigger	Label	Meter
/metabuilder/clear_log	queued	task			
/metabuilder/regenerate	complete	task		date: Tue Jun 28 16:36:50 GMT 2016	
/metabuilder/remove_build_dirs	complete	task			
/metview	aborted	suite		branch: 4.7.1	
/metview/ecgb	suspended	family			
/metview/ecgb/build	submitted	task	config == complete	host:	<input type="text"/> progress: 0
/metview/ecgb/config	queued	task	make_build_dir == complete	version:	
/metview/ecgb/git_srcs	queued	task	var_summary == complete	branch:	
/metview/ecgb/git_update_repos	queued	task	git_srcs == complete		
/metview/ecgb/install	suspended	task	set_write_permissions == complete		
/metview/ecgb/install_log	queued	task	install_test == complete		

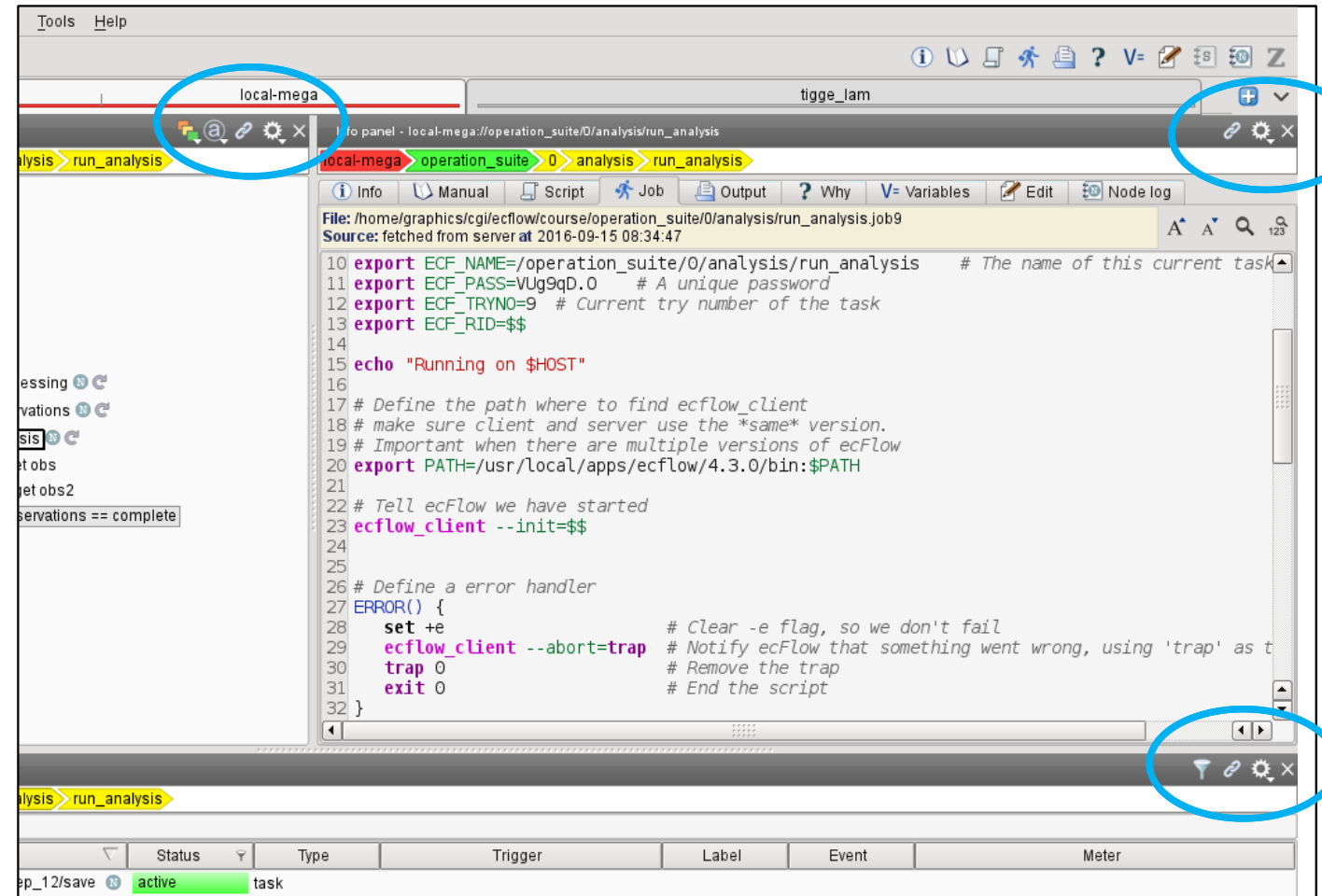
Interaction between panels

- By default, selecting node in one panel causes it to be selected in the other panels (in the same tab)
- Click the 'link' icon in each panel header to change whether it reacts to selections in other panels

• Linked = listening



• Unlinked = not listening



```
File: /home/graphics/cgi/ecflow/course/operation_suite/0/analysis/run_analysis.job9
Source: fetched from server at 2016-09-15 08:34:47

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 }
```

Status	Type	Trigger	Label	Event	Meter
active	task				

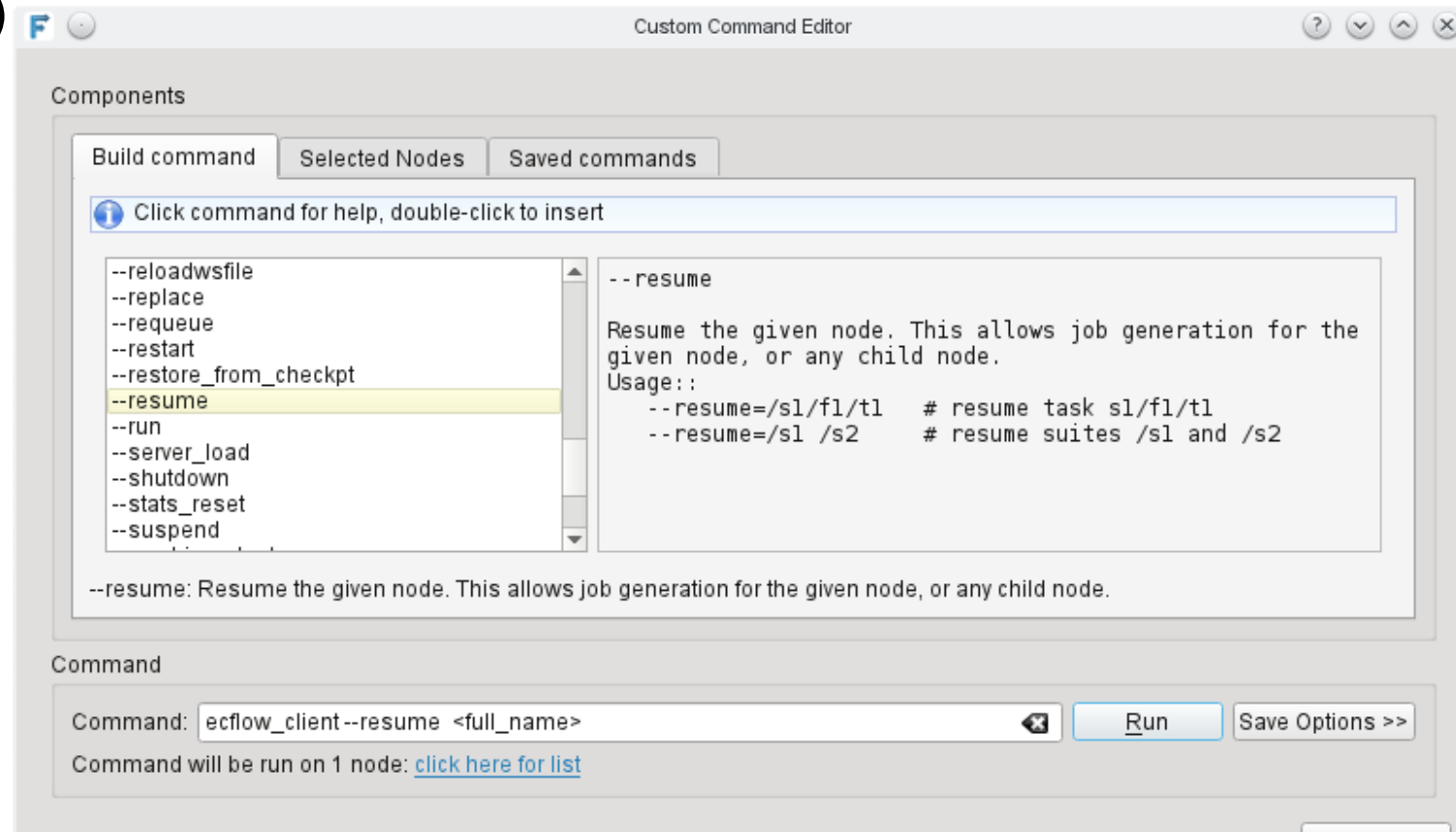
Custom commands

- The node context menu provides node-sensitive commands
- If you know the ecFlow commands, you can add your own
- Context menu -> Custom... -> Manage commands...

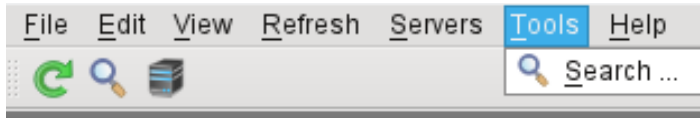
The image shows two overlapping windows from the metview application. The top window is the context menu for a node named 'metview'. It contains the following items: Suspend (highlighted), Requeue, Requeue aborted, Free deps, Defstatus, Force, Order, Special, Begin, Cancel, Custom, Search..., Info..., Variables..., Copy node, Expand all, and Collapse all. The 'Custom' option is selected, opening a second window titled 'metview' with a wrench icon and the text 'Manage commands...'. This window lists several custom commands: delete_yes, Recent, ecflow_client --order <full_name> bottom, ecflow_client --delete <full_name>, ecflow_client --debug <full_name>, and ecflow_client --terminate <full_name>.

Custom commands (2)

- The **Build command** dialogue helps you create a command
- You can run it once (it will appear in the Custom -> Recent menu) or save it with a name



Searching for nodes



- Powerful interface to find nodes
- Results show 'live' status of nodes
- Can search for nodes and attributes
- Click on a search result to select it in the linked panels

Nodes | Attributes*

Types ↶ Attribute-specific options

- date
- event
- label
- limit
- limiter
- meter
- repeat
- time
- trigger
- variable

Meter name:

Meter value:

Trigger expression:

Scope

Search in servers: ↶

Search root node:

Global options

Max results: ↶

Case sensitive

Search

Show:

Editor

Nodes* | Attributes

Name:

Path: ↶

Type ↶ Status ↶ Flag ↶

<input type="checkbox"/> server	<input checked="" type="checkbox"/> aborted	<input type="checkbox"/> is_late
<input type="checkbox"/> suite	<input type="checkbox"/> active	<input type="checkbox"/> has_date
<input checked="" type="checkbox"/> family	<input checked="" type="checkbox"/> complete	<input type="checkbox"/> has_message
<input checked="" type="checkbox"/> task	<input checked="" type="checkbox"/> queued	<input type="checkbox"/> has_time
<input type="checkbox"/> alias	<input type="checkbox"/> submitted	<input type="checkbox"/> is_rerun
	<input type="checkbox"/> suspended	<input type="checkbox"/> is_waiting
	<input type="checkbox"/> unknown	<input type="checkbox"/> is_zombie

Query

```
SELECT node FROM ecflow-metab WHERE (node_path = '*emos*') and (family or task) and (aborted or complete or queued) LIMIT 50000
```

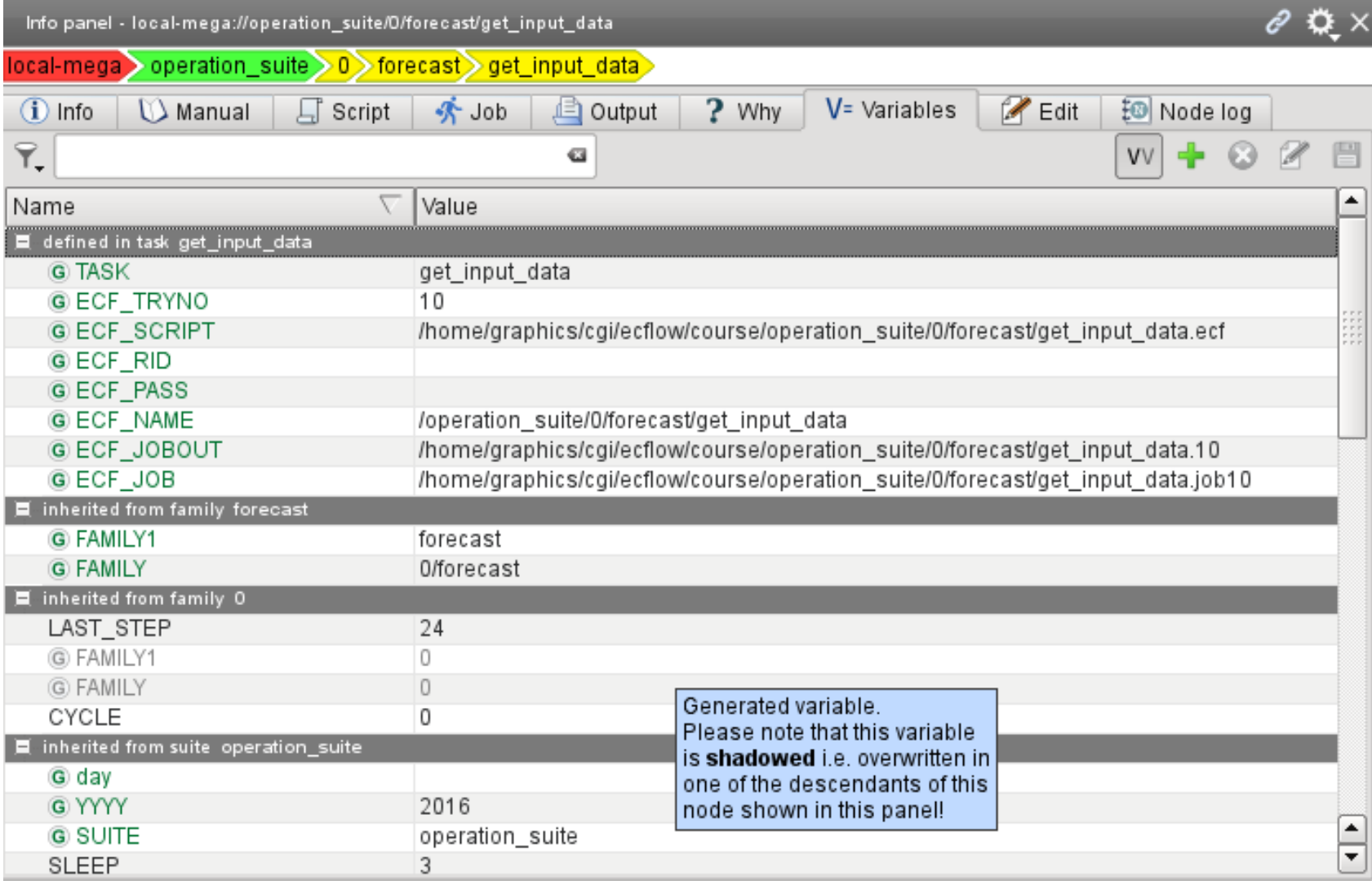
Server	Node	State	Type
ecflow-metab	/libemos/opensuse131/gnu.61/var_si	aborted	task
ecflow-metab	/libemos/opensuse131/gnu.61/test	queued	task
ecflow-metab	/libemos/opensuse131/gnu.61/remov	complete	task
ecflow-metab	/libemos/opensuse131/gnu.61/modu	complete	task
ecflow-metab	/libemos/opensuse131/gnu.61/make_	queued	task
ecflow-metab	/libemos/opensuse131/gnu.61/install	queued	task
ecflow-metab	/libemos/opensuse131/gnu.61/git_sr	queued	task
ecflow-metab	/libemos/opensuse131/gnu.61/config	queued	task
ecflow-metab	/libemos/opensuse131/gnu.61/build	queued	task
ecflow-metab	/libemos/opensuse131/gnu.61/add_v	complete	task
ecflow-metab	/libemos/opensuse131/gnu.61	aborted	family
ecflow-metab	/libemos/opensuse131/gnu.53/wipe_	complete	task
ecflow-metab	/libemos/opensuse131/gnu.53/var_si	complete	task
ecflow-metab	/libemos/opensuse131/gnu.53/test	complete	task

7085 items found in 0.2 s

Close

Info panel

- Node specific
- Displays node attributes, logs, outputs, variables



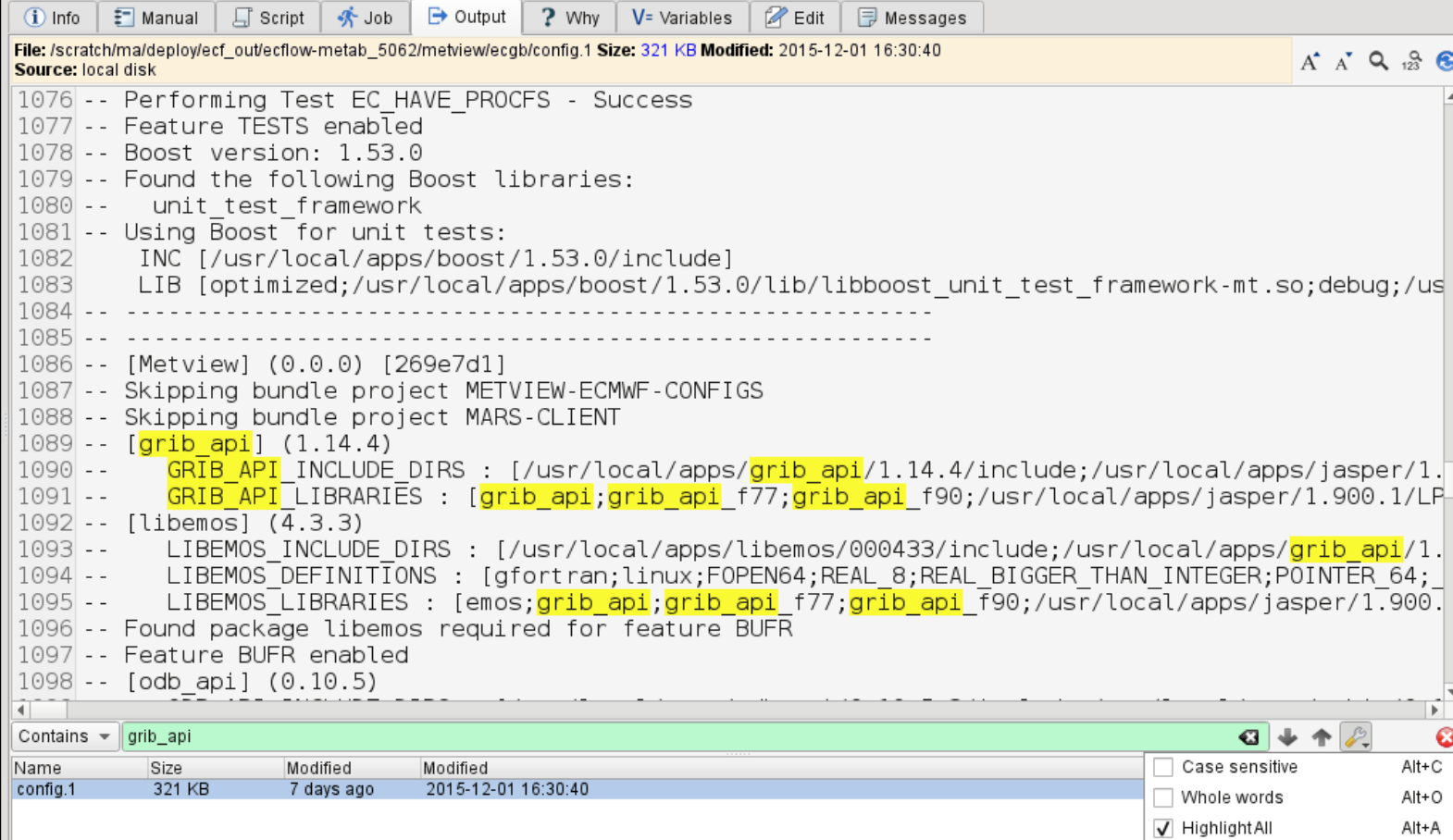
The screenshot shows the 'Info panel' for a node named 'get_input_data' within the 'forecast' family of the 'operation_suite/0' node. The interface includes a breadcrumb trail, a menu bar with options like 'Info', 'Manual', 'Script', 'Job', 'Output', 'Why', 'Variables', 'Edit', and 'Node log', and a search bar. The main content is a table with 'Name' and 'Value' columns, organized into sections: 'defined in task get_input_data', 'inherited from family forecast', 'inherited from family 0', and 'inherited from suite operation_suite'. A blue callout box highlights the 'CYCLE' variable, stating it is a generated variable that is shadowed (overwritten) in a descendant node.

Name	Value
defined in task get_input_data	
TASK	get_input_data
ECF_TRYNO	10
ECF_SCRIPT	/home/graphics/cgi/ecflow/course/operation_suite/0/forecast/get_input_data.ecf
ECF_RID	
ECF_PASS	
ECF_NAME	/operation_suite/0/forecast/get_input_data
ECF_JOBOUT	/home/graphics/cgi/ecflow/course/operation_suite/0/forecast/get_input_data.10
ECF_JOB	/home/graphics/cgi/ecflow/course/operation_suite/0/forecast/get_input_data.job10
inherited from family forecast	
FAMILY1	forecast
FAMILY	0/forecast
inherited from family 0	
LAST_STEP	24
FAMILY1	0
FAMILY	0
CYCLE	0
inherited from suite operation_suite	
day	
YYYY	2016
SUITE	operation_suite
SLEEP	3

Generated variable.
Please note that this variable is **shadowed** i.e. overwritten in one of the descendants of this node shown in this panel!

Job output

- Flexible searching (regular expressions, etc.)
- Can define a default search which takes place whenever you refresh the output
- Handles large output files (e.g. >1GB) without much memory overhead



```
File: /scratch/ma/deploy/ecf_out/ecflow-metab_5062/metview/ecgb/config.1 Size: 321 KB Modified: 2015-12-01 16:30:40
Source: local disk
1076 -- Performing Test EC_HAVE_PROCFCS - Success
1077 -- Feature TESTS enabled
1078 -- Boost version: 1.53.0
1079 -- Found the following Boost libraries:
1080 --   unit_test_framework
1081 -- Using Boost for unit tests:
1082   INC [/usr/local/apps/boost/1.53.0/include]
1083   LIB [optimized;/usr/local/apps/boost/1.53.0/lib/libboost_unit_test_framework-mt.so;debug;/usr
1084 ---
1085 ---
1086 -- [Metview] (0.0.0) [269e7d1]
1087 -- Skipping bundle project METVIEW-ECMWF-CONFIGS
1088 -- Skipping bundle project MARS-CLIENT
1089 -- [grib_api] (1.14.4)
1090 --   GRIB_API_INCLUDE_DIRS : [/usr/local/apps/grib_api/1.14.4/include;/usr/local/apps/jasper/1.
1091 --   GRIB_API_LIBRARIES : [grib_api;grib_api_f77;grib_api_f90;/usr/local/apps/jasper/1.900.1/LP
1092 -- [libemos] (4.3.3)
1093 --   LIBEMOS_INCLUDE_DIRS : [/usr/local/apps/libemos/000433/include;/usr/local/apps/grib_api/1.
1094 --   LIBEMOS_DEFINITIONS : [gfortran;linux;FOPEN64;REAL_8;REAL_BIGGER_THAN_INTEGER;POINTER_64;
1095 --   LIBEMOS_LIBRARIES : [emos;grib_api;grib_api_f77;grib_api_f90;/usr/local/apps/jasper/1.900.
1096 -- Found package libemos required for feature BUFR
1097 -- Feature BUFR enabled
1098 -- [odb_api] (0.10.5)
```

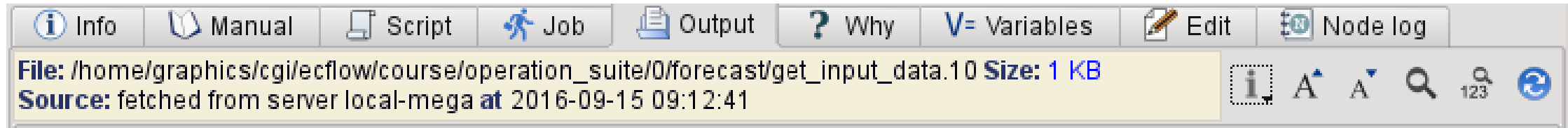
Name	Size	Modified	Modified
config.1	321 KB	7 days ago	2015-12-01 16:30:40

Contains: grib_api

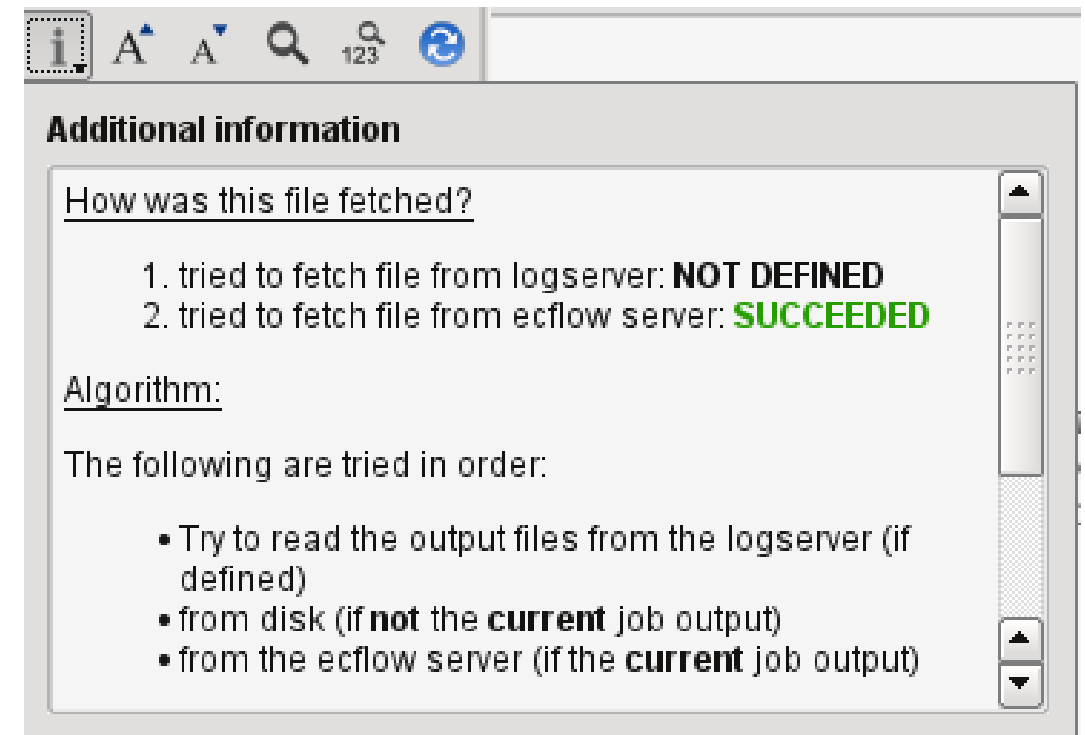
Case sensitive Alt+C
 Whole words Alt+O
 HighlightAll Alt+A

Job output

- Be transparent in what ecFlowUI does
- Example: where did we find the log file?



The screenshot shows the top part of the ecFlowUI interface. It features a menu bar with buttons for 'Info', 'Manual', 'Script', 'Job', 'Output', 'Why', 'Variables', 'Edit', and 'Node log'. Below the menu bar, a yellow highlighted area displays file information: 'File: /home/graphics/cgi/ecflow/course/operation_suite/0/forecast/get_input_data.10 Size: 1 KB' and 'Source: fetched from server local-mega at 2016-09-15 09:12:41'. To the right of this text are several icons, including an information icon, zoom in and out arrows, a search icon, and a refresh icon.



The screenshot shows a dialog box titled 'Additional information'. It contains the following text:

How was this file fetched?

1. tried to fetch file from logserver: **NOT DEFINED**
2. tried to fetch file from ecflow server: **SUCCEEDED**

Algorithm:

The following are tried in order:

- Try to read the output files from the logserver (if defined)
- from disk (if **not** the **current** job output)
- from the ecflow server (if the **current** job output)

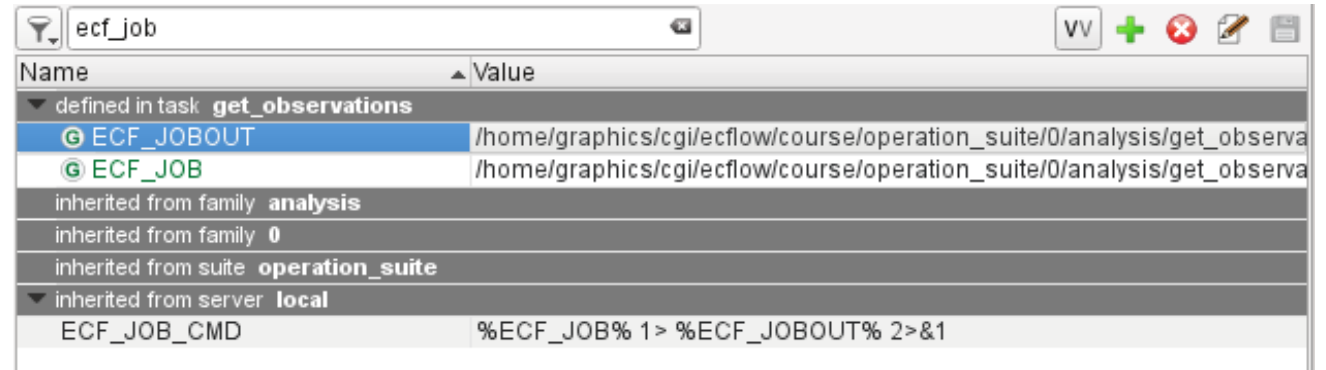
Variables

- Info Panel -> Variables
- Shows all the variables that apply to the selected node
- Some variables are inherited from ancestors in the tree
- Some variables are 'generated', i.e. not defined in the suite but created by ecFlow
- Some variables are read-only

Name	Value
defined in task: get_observations	
TASK	get_observationsxx
MM	08
ECF_URL	wiki/display/ECFLOW/Home/
ECF_TRYNO	0
ECF_SCRIPT	/home/graphics/cgi/ecflow/course/operation_suite/0/analysis/get_obs
ECF_RID	
ECF_PASS	FREE
ECF_NAME	/operation_suite/0/analysis/get_observations
ECF_JOBOUT	/home/graphics/cgi/ecflow/course/operation_suite/0/analysis/get_obs
ECF_JOB	/home/graphics/cgi/ecflow/course/operation_suite/0/analysis/get_obs
inherited from family: analysis	
FAMILY1	analysis
FAMILY	0/analysis
inherited from family: 0	
LAST_STEP	24
CYCLE	0
inherited from suite: operation_suite	
day	
YYYY	2016
SUITE	operation_suite
SLEEP	6
MONTH	june
ECF_TIME	14:30
ECF_LOGPORT	9316
ECF_LOGHOST	
ECF_INCLUDE	/home/graphics/cgi/ecflow/course
ECF_HOME	/home/graphics/cgi/ecflow/course
ECF_FILES	/home/graphics/cgi/ecflow/course/oper
ECF_DATE	20160608
ECF_CLOCK	wednesday:june:3:160
DOY	160
DOW	3
DD	08
DAY	wednesday
DATE	08.06.2016
inherited from server: local	
ECF_VERSION	4.1.0
ECF_URL_CMD	\${BROWSER:=firefox} -remote 'openURL(%ECF_URL_BASE%/%ECF_
ECF_URL_BASE	https://software.ecmwf.int/
ECF_TRIES	4
ECF_STATUS_CMD	ps --sid %ECF_RID%
ECF_PORT	16755
ECF_PID	4131
ECF_NODE	odorir
ECF_MICRO	%
ECF_LOG	/home/graphics/cgi/ecflow_server/odorir.16755.ecf.log
ECF_LISTS	/home/graphics/cgi/ecflow_server/ecf.lists

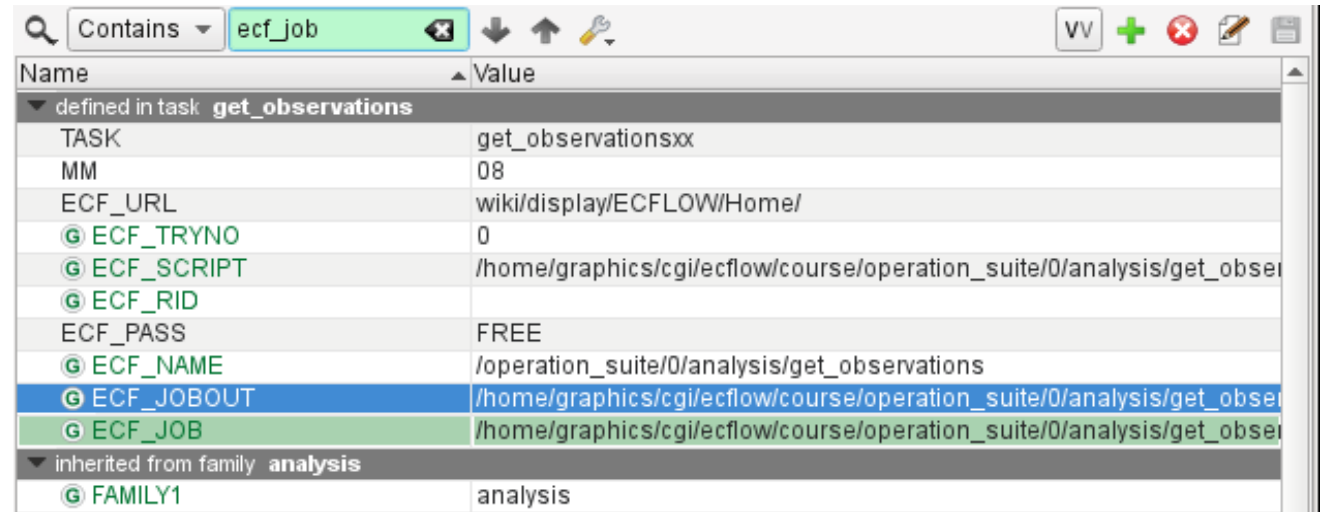
Variables - Searching

- Can filter the list of variables
- Can search for variables



The screenshot shows a search interface with a search bar containing 'ecf_job'. The results are displayed in a table with columns 'Name' and 'Value'. The search results are filtered to show only variables defined in the task 'get_observations'.

Name	Value
▼ defined in task get_observations	
ECF_JOBOUT	/home/graphics/cgi/ecflow/course/operation_suite/0/analysis/get_observa
ECF_JOB	/home/graphics/cgi/ecflow/course/operation_suite/0/analysis/get_observa
inherited from family analysis	
inherited from family 0	
inherited from suite operation_suite	
▼ inherited from server local	
ECF_JOB_CMD	%ECF_JOB% 1> %ECF_JOBOUT% 2>&1

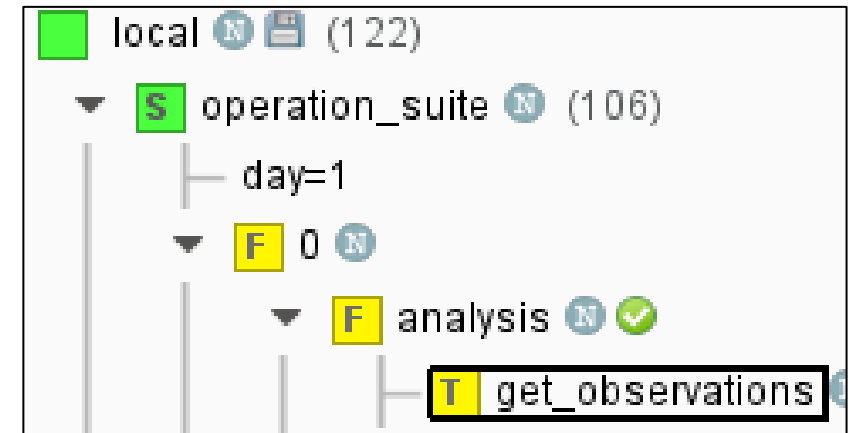


The screenshot shows a search interface with a search bar containing 'ecf_job' and a filter dropdown set to 'Contains'. The results are displayed in a table with columns 'Name' and 'Value'. The search results are filtered to show variables containing 'ecf_job'.

Name	Value
▼ defined in task get_observations	
TASK	get_observationsxx
MM	08
ECF_URL	wiki/display/ECFLOW/Home/
ECF_TRYNO	0
ECF_SCRIPT	/home/graphics/cgi/ecflow/course/operation_suite/0/analysis/get_observa
ECF_RID	
ECF_PASS	FREE
ECF_NAME	/operation_suite/0/analysis/get_observations
ECF_JOBOUT	/home/graphics/cgi/ecflow/course/operation_suite/0/analysis/get_observa
ECF_JOB	/home/graphics/cgi/ecflow/course/operation_suite/0/analysis/get_observa
▼ inherited from family analysis	
FAMILY1	analysis

Variables - Shadowing

- Variables redefined deeper in the hierarchy are “shadowed”
- Can choose to display them (they will be in grey) or not (use the ‘VV’ button)



Name	Value
ECF_JOB	/home/graphics/cgi/ecflow/course/operation_suite/0/analysis/get_obs
inherited from family analysis	
FAMILY1	analysis
FAMILY	0/analysis
inherited from family 0	
LAST_STEP	24
FAMILY1	0
FAMILY	0
CYCLE	0
inherited from suite operation_suite	

Current status

- Test Version available for download (or run on ecgate)
- ecFlowUI is still in development but is used daily by many users
- Already widely used at ECMWF, but not an 'official' replacement for ecfview yet; with upcoming release will roll out to more users
- We are happy to receive feedback and feature requests

Questions?

Practical

- Create the layout as shown: Tree View + Info Panel + Table; apply a status filter to the Table View so that it only shows Active or Aborted tasks; try a more complex filter
- **Suspend** some nodes, then **Search** for suspended nodes and **Resume** them from the search results
- Experiment with the ‘linking’ behaviour
- Right-click on a node and select “Output” to get its output in a new window
- Try the icons in the Tree View’s panel header
- Look at the Variables tab – try changing some variables, including variables that belong to a parent node – does the behaviour make sense?
- Explore settings in Edit -> Preferences menu

The screenshot displays the ecFlowUI (4.5.0) interface. The Tree View on the left shows a hierarchy of nodes, including 'metabuilder' (4987) and 'mdev' (795). The Info panel on the right shows the source file path and the output of a script, which includes a Python script snippet. The Table view at the bottom shows a list of nodes with columns for Node, Status, Type, Trigger, Label, Event, Meter, and Status changed. The filter is set to 'SELECT node WHERE (aborted or active)'. The table contains several rows of data, including nodes with status 'aborted' and 'active'.

Node	Status	Type	Trigger	Label	Event	Meter	Status changed
/eccodes/leap42/gnu.53/	aborted	task	install_check == complete				2017-Jan-05 17:19:00
/eccodes/leap42/gnu.61	aborted	family					2017-Jan-05 17:19:00
/eccodes/leap42/gnu.61/	aborted	task	install_check == complete				2017-Jan-05 17:19:00
/eccodes/leap42/pgi	aborted	family					2017-Jan-05 17:19:00
/eccodes/leap42/pgi/inst	aborted	task	install_check == complete				2017-Jan-05 17:19:00
/atlas_bundle	aborted	suite		branch: 0.8.2.0			2016-Dec-09 16:37:00
/atlas_bundle/ixg	aborted	family					2016-Dec-09 16:37:00
/atlas_bundle/ixg/gnu.53	aborted	family					2016-Dec-09 16:37:00

