

Advanced Tools Window - Test Runner UI

Advanced Tools

Test Runner

Git Control

System Console

Server Workspace

File Tree

Terminal

lively.tests.TestFrameworkTests

Load Module

▶ TestSuiteTest

▼ lively

▼ tests

▼ TestsFrameworkTests

▶ AsyncTestCaseTest

▶ MockTest

▶ NativeStackTest

▶ TestCaseTest

▶ TestResultTest

62 run, 0 failed




Reset

Browse

Brench

Run

Advanced Tools Window - System Console UI

 Advanced Tools  

Test Runner

Git Control

System Console

Server Workspace

File Tree

Terminal

Log Messages

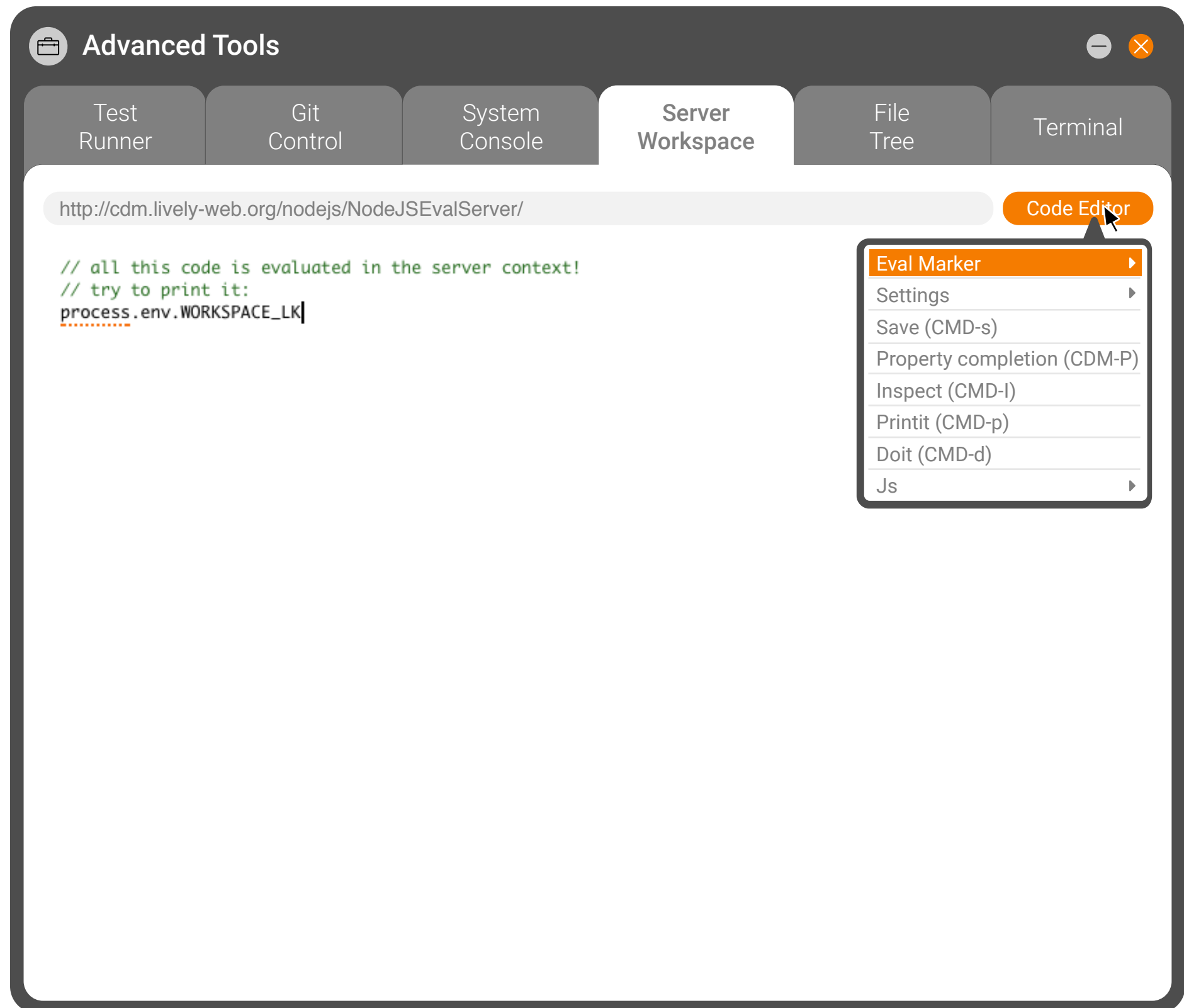
```
[log] http://cdm.lively-web.org/core/anonymous_module_32 loaded in 1 ms
[log] http://cdm.lively-web.org/core/anonymous_module_31 loaded in 96 ms
[log] http://cdm.lively-web.org/core/anonymous_module_34 loaded in 0 ms
[log] http://cdm.lively-web.org/core/anonymous_module_35 loaded in 2365 ms
[log] http://cdm.lively-web.org/core/anonymous_module_36 loaded in 1 ms
|
```

Clear

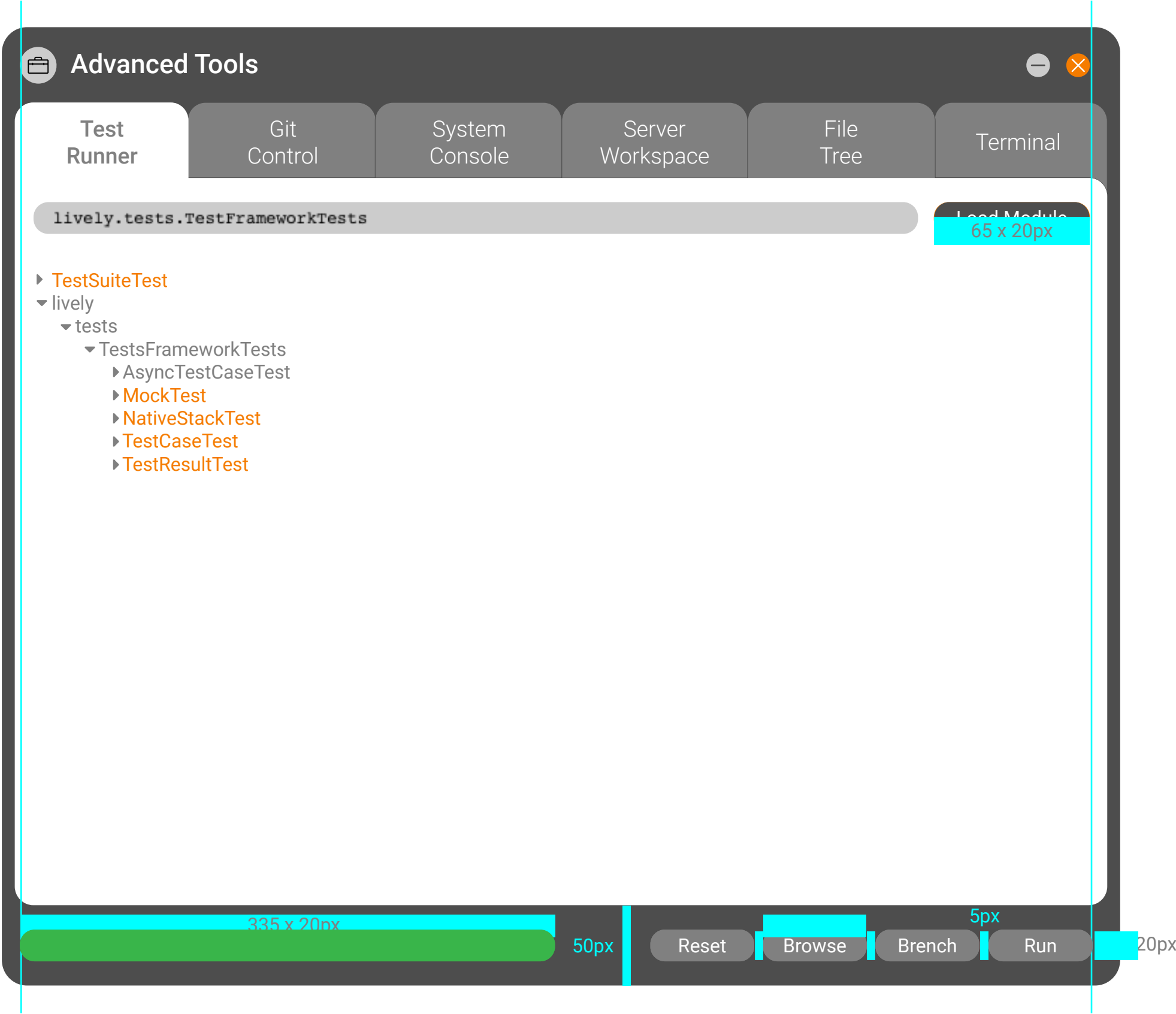
Reattach

☒ Follow Log

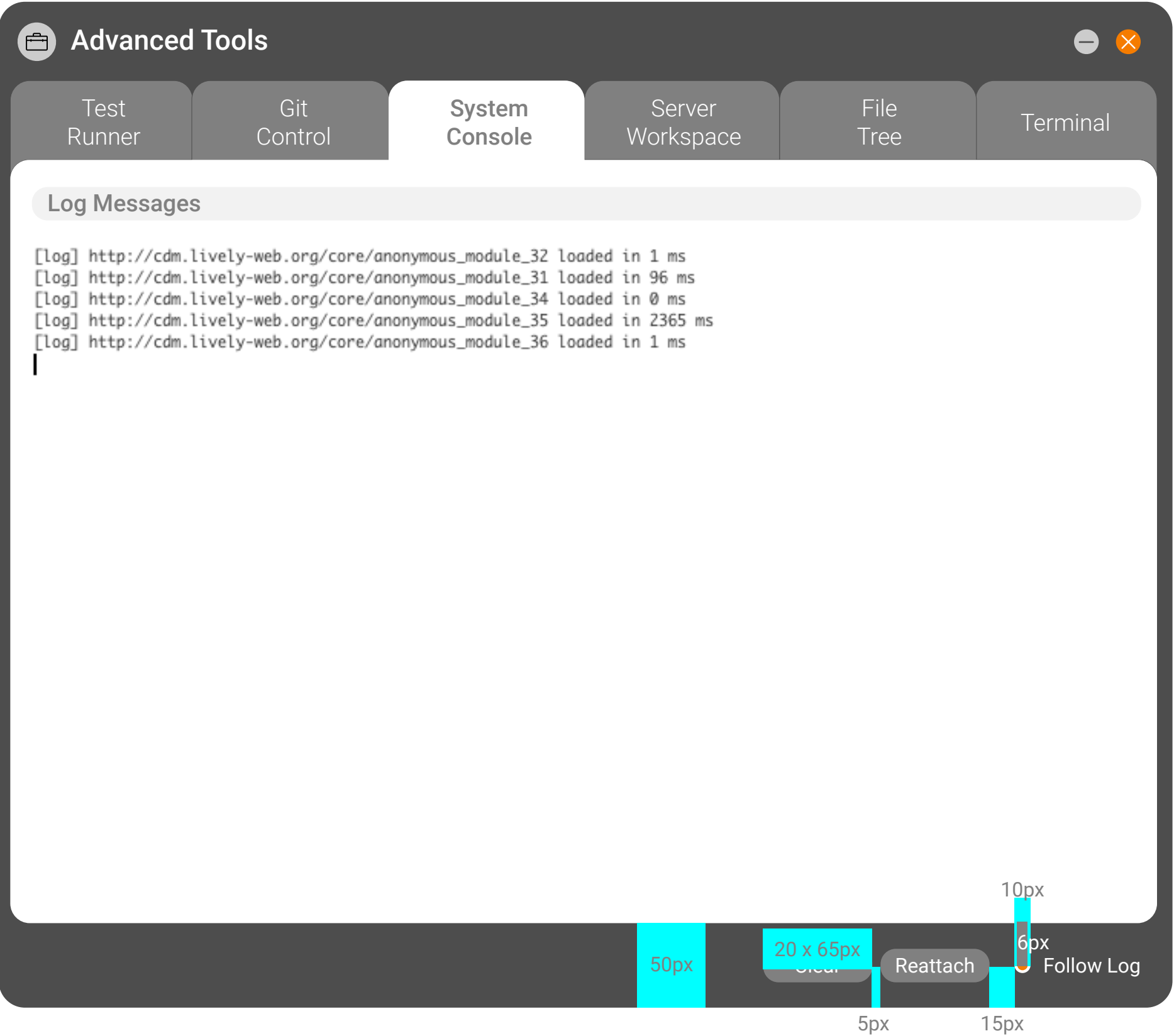
Advanced Tools Window - Server Workspace Console UI



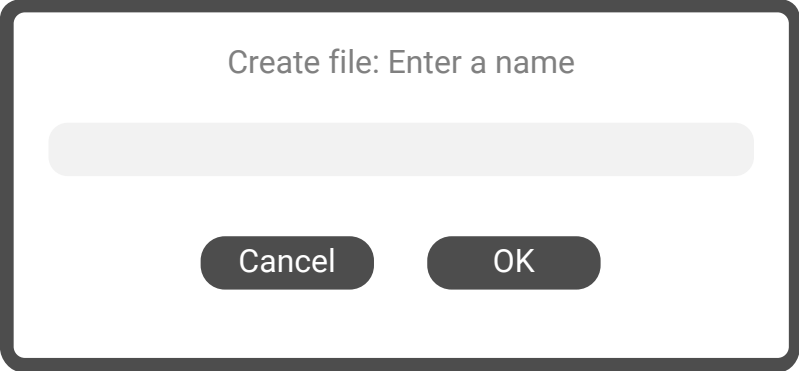
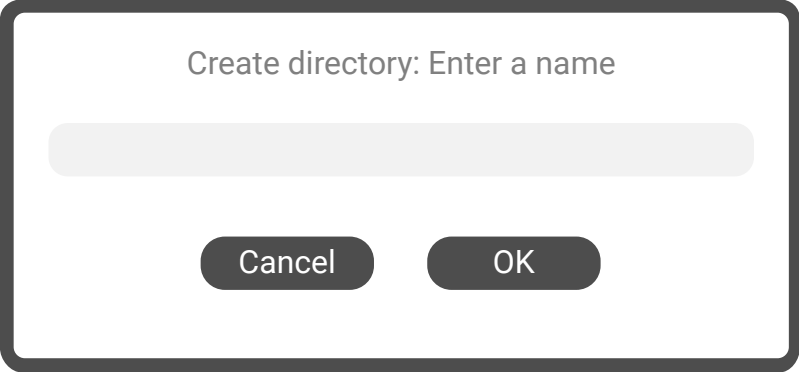
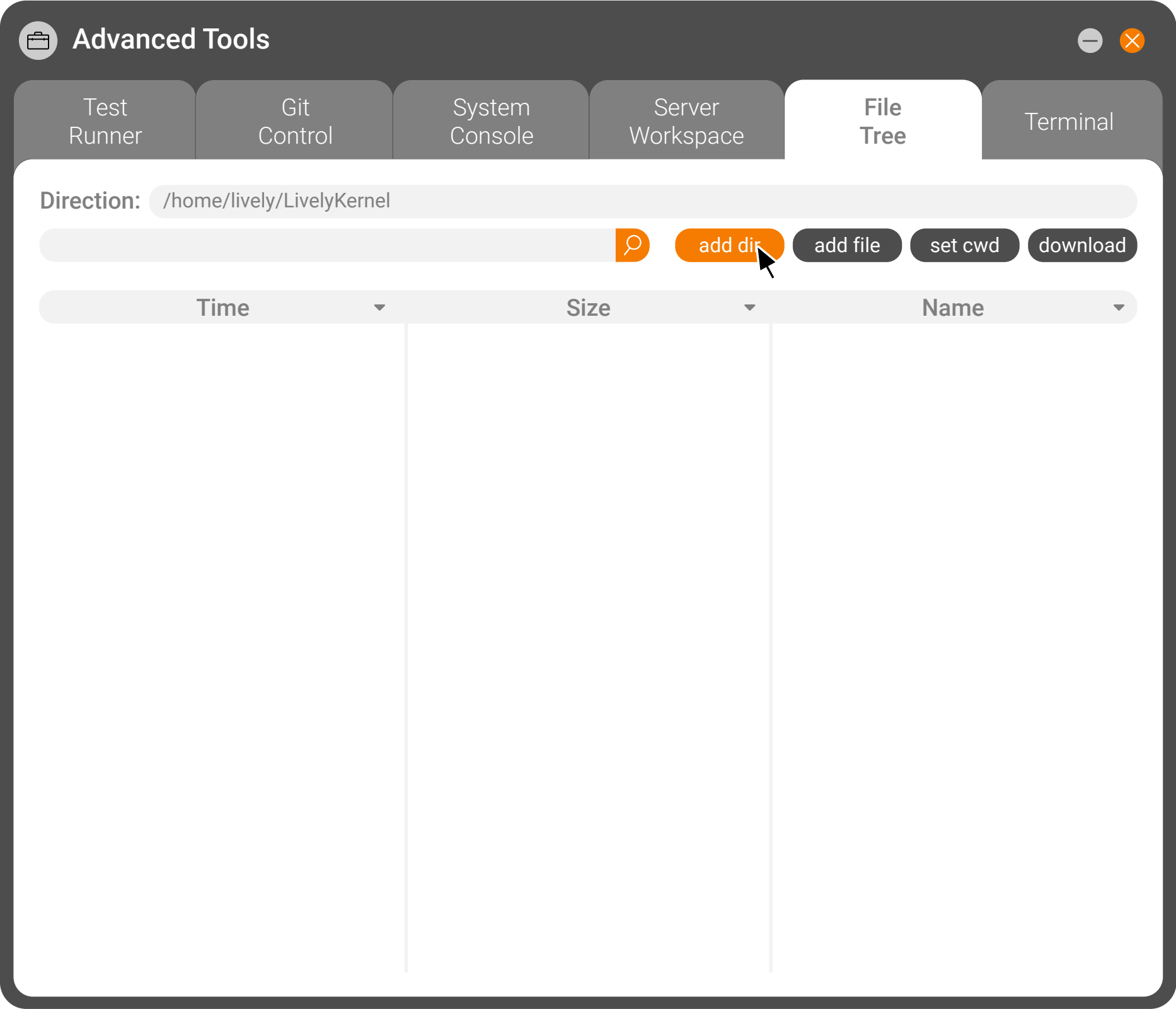
Advanced Tools Window - Test Runner UI



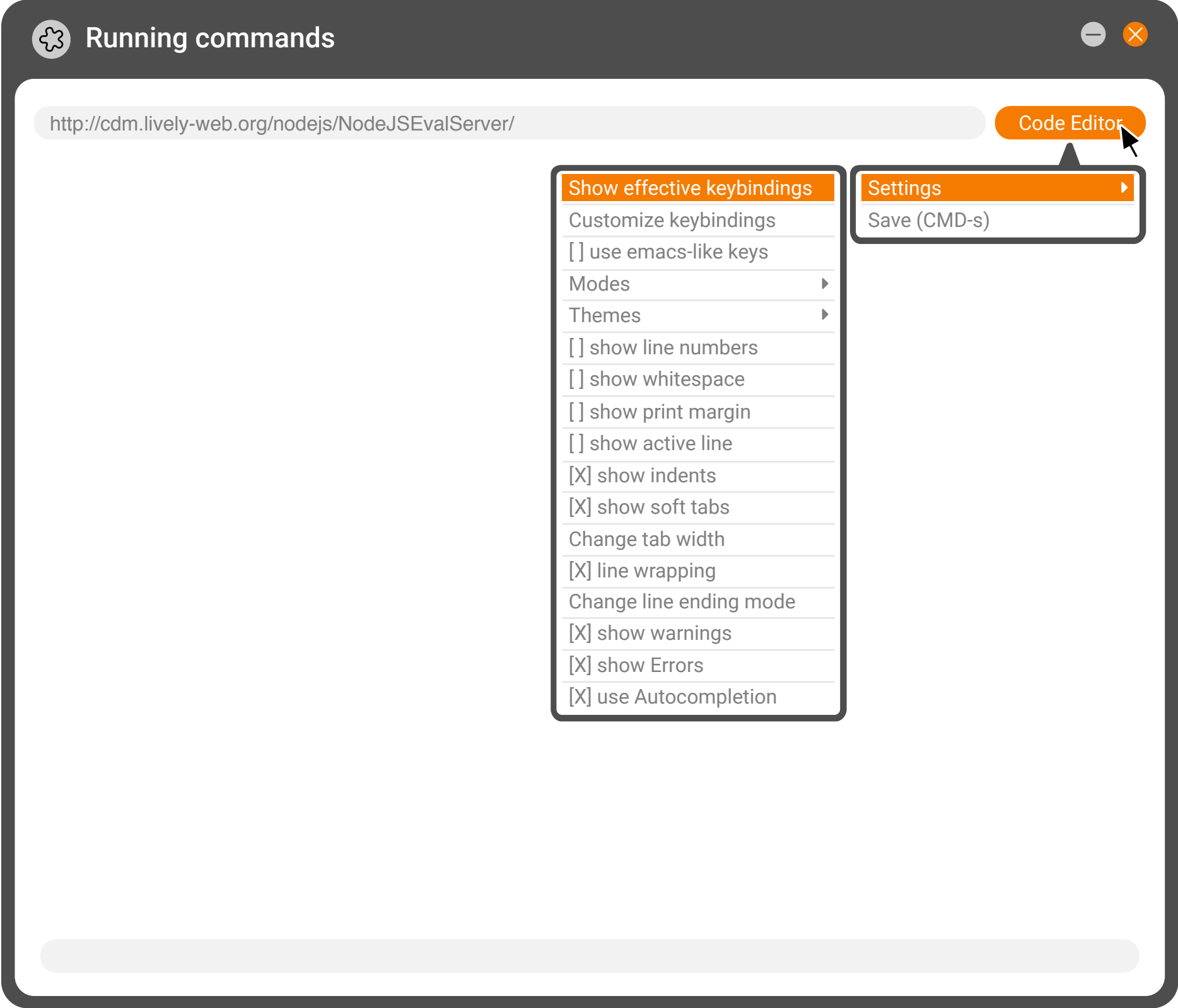
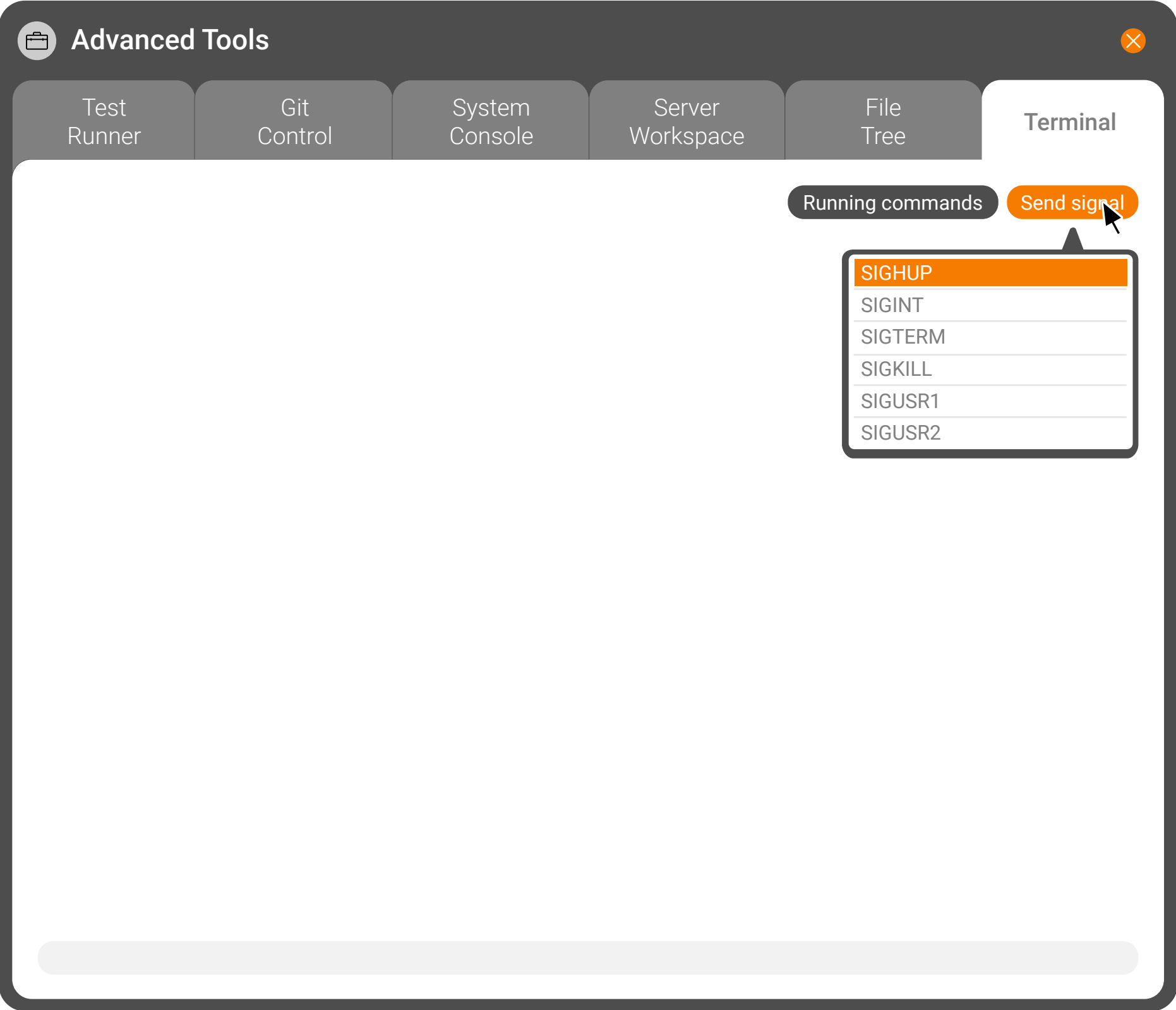
Advanced Tools Window - System Console UI



Advanced Tools Window - File Tree UI



Advanced Tools Window - Terminal UI



Advanced Tools Window - Terminal UI

Advanced Tools

Test Runner

Git Control

System Console

Server Workspace

File Tree

Terminal

Status

Log

Console

Documentation

Wip

Working directory

Onto cdm

remote: / -

local: cdm - null

head: 07d1927 Changed the if condition for the morph move undo due to a core lively issue. Now explicitly checking th

Stashes

0: WIP on cdm: 50bf296 fixed the undo issue related to grabbing, moving, submorphs.... again, but properly this time!

Staged changes:

modified: core/lively/PartsBin.js

modified: core/lively/data/ImageUpload.js

modified: core/lively/morphic/Rendering.js

modified: core/servers/QBFScoresServer.js

modified: favicon.ico

modified: welcome.html

Untracked files:

02_logo-34.jpg

Default/

Fun/

QBFScores.json

Untitled-1.png

admin/

core/lively/morphic/.Halos.js.swp

core/servers/AnonymousWormHole.js

core/servers/IndexPageServer.js

demos/

login

questions/

test/

tutorials/

wireframe

wireframeflow101

Advanced Tools

Test Runner

Git Control

System Console

Server Workspace

File Tree

Terminal

Status

Log

Console

Documentation

Wip

Working directory

* 07d1927 - (HEAD, cdm) Changed the if condition for the morph move undo due to a core lively issue. Now explicitly checking that the targetMorph is not being Grabbed before adjusting origin coordinates. (Riz Panjwani, 5 hours ago)

| * dafca08 - (refs/stash) WIP on cdm: 50bf296 fixed the undo issue related to grabbing, moving, submorphs.... again, but properly this time! (lively-web, 4 hours ago)

| / /

| * 3e9cbbd - index on cdm: 50bf296 fixed the undo issue related to grabbing, moving, submorphs... again, but properly this time! (lively-web, 4 hours ago)

| /

| * 50bf296 - fixed the undo issue related to grabbing, moving, submorphs.... again, but properly this time! (Riz Panjwani, 3 days ago)

| * 9a5e47b - Revert "fixed undo bug related to Grab actions where it was registering the action twice and causing problems." (Riz Panjwani, 3 days ago)

| * 8e5e6b9 - Revert "fixed the issue with undoing a morph moving. When moving a morph, it incorrectly recorded the coordinates of where the morph origin was at the time mouseMove was triggered rather than its true start origin." (Riz Panjwani, 3 days ago)

| * ad8a761 - fixed typo: changed Console to console in amendMorphicAction. also added a null check for evt.hand in logTransformationForUndo. (Riz Panjwani, 7 days ago)

| * 1fa6dbc - fixed the issue with undoing a morph moving. When moving a morph, it incorrectly recorded the coordinates of where the morph origin was at the time mouseMove was triggered rather than its true start origin. (Riz Panjwani, 8 days ago)

| * e454e6d - Increased max undo queue size from 7 to 20 actions. (root, 3 weeks ago)

| * add83ab - fixed undo bug related to Grab actions where it was registering the action twice and causing problems. (root, 3 weeks ago)

| * aa027d0 - implemented shortcuts for undo/redo functionality (root, 4 weeks ago)

| * cbe74bd - Added Lively Help System (Matth, 4 weeks ago)

| * d75b0f3 - UndoRedo: Another this.owner fix (Marko Röder, 5 weeks ago)

| * 27239b4 - UndoRedo: Fixing reference to morph (not targetMorph) (Marko Röder, 6 weeks ago)

| * 314873d - Dan's UndoRedo work for morphic halos and simple interaction (Marko Röder, 6 weeks ago)

| * bdf062d - (origin/morphic-undoredo) Changed the if condition for the morph move undo due to a core lively issue. Now explicitly checking that the targetMorph is not being Grabbed before adjusting origin coordinates. (Riz Panjwani, 5 hours ago)

Advanced Tools

Test Runner

Git Control

System Console

Server Workspace

File Tree

Terminal

Status

Log

Console

Documentation

Wip

Working directory

edit.git/config

edit.gitignore

Advanced Tools

Test Runner

Git Control

System Console

Server Workspace

File Tree

Terminal

Status

Log

Console

Documentation

Wip

Working directory

All Lively installations, such as you might have on localhost, or such as the lively-kernel.org/core installation, are separate instances of the master installation on Git. The GitControl tool makes it easy to update an installation and to commit changes made in that installation up to the master on Git.

General commit and update workflow

As a general rule, prior to making a bunch of changes, it is good to use the GitControl to do a "pull", bringing your installation up to date so that the only changes you will have to deal with be those involved in your project.

Suppose you have changed a method in Widgets.js. When you open the GitControl, this file will appear as an "unstaged change", or you may have several files listed there that you have changed. You can use the "diff" button to see the changes that will be committed. [this does not work for me but it should for you. The alternative is to go tot the console window and type "git diff" or "git diff " followed by the pathname of a specific file]

Maybe we need some discussion about what to do if things appear in the diff that you do not want. The simplest answer here would go to go fix the file.

In preparation to commit, you need all your local changes to be either "staged" (in preparatin to commit), "stashed" (set aside from the commit but not forgotten. perhaps as you have more to do before committing), or forgotten (the X), meaning the changes will be lost, though you may still have that file around. So do one of these three things to all your unstaged changes.

Now you are ready to commit your (now) staged changes, and this is done by pressing the "commit" button. Here you will be asked for a short comment about what these changes are. This will appear in the log which can be seen under the "log" tab.

Next you should press "pull". This may take a little while and should then print a few lines ending with a pair of version numbers on the master. [for some reason my system asks me for my name and password at this point which can be entered in the terminal window]

Advanced Tools

Test Runner

Git Control

System Console

Server Workspace

File Tree

Terminal

Status

Log

Console

Documentation

Wip

Working directory

Quick Publish

Workspace

Todo List

1 that.setPosition(pt(0,0))

2 this.owner.owner.addMorph(that)

3

4 this.owner.owner.getTabContainer().getTabBar().rearrangeTabs()

5 t = this.owner.owner.getTabContainer().getTabBar().tabs.pop()

6 t2 = this.owner.owner.getTabContainer().getTabBar().tabs.pop()

7

8 this.owner.owner.getTabContainer().getTabBar().getTabs

1 PRETTY FORMATS

2 If the commit is a merge, and if the pretty-format is not oneline,

3 email or raw, an additional line is inserted before the Author: line.

4 This line begins with "Merge: " and the sha1s of ancestral commits are

5 printed, separated by spaces. Note that the listed commits may not

6 necessarily be the list of the direct parent commits if you have

7 limited your view of history: for example, if you are only interested

8 in changes related to a certain directory or file.

9

10 There are several built-in formats, and you can define additional

11 formats by setting a pretty.<name> config option to either another

12 format name, or a format: string, as described below (see git-

13 config(1)). Here are the details of the built-in formats:

14

15 o oneline

16

17 <sha1> <title line>

Advanced Tools

Test Runner

Git Control

System Console

Server Workspace

File Tree

Terminal

Status

Log

Console

Documentation

Wip

Working directory

Quick Publish

Workspace

Todo List

Reload Browser

Close Window

Save

Open Workspace

Open Object Editor

Open System Browser

Reload Browser

Close Window

Save

Open Workspace

Open Object Editor

Open System Browser

Reload Browser

Close Window

Save

Advanced Tools

Test Runner

Git Control

System Console

Server Workspace

File Tree

Terminal

Status

Log

Console

Documentation

Wip

Working directory

Set dir

⌂

Default

Fun

PartsBin

admin

apps

benchmarks

bin

core

demos

projects

questions

test

tutorials

users

voltaire

.gitattributes