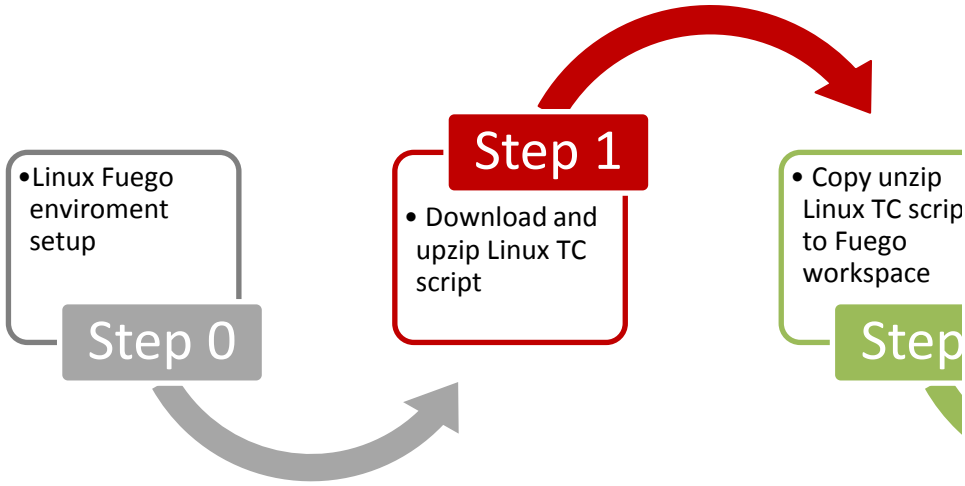
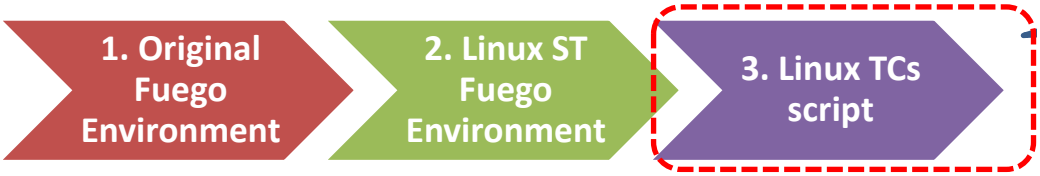
GUIDELINE TO EMBED TCs INTO FUEGO ENVIRONMENT

Because this document will be used internally in REL - RVC, some of reference links are inte



Step 0 Setup Linux FUEGO Environment

To embed an Linux TC to FUEGO environment, user has to setup an Linux FUEGO Environ



1. Download original FUEGO Environment from: https://bitbucket.org/cogentembedde
Refer to Implementation_Spec_TAF_LinuxST_vX.X.doc (part 1.4) to setup the environment

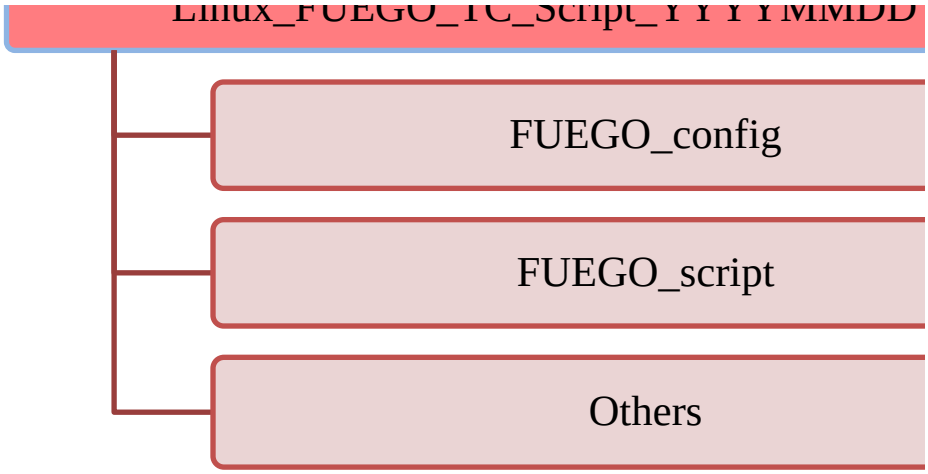
2. Download Linux FUEGO Environment from Redmine.
Refer to Implementation_Spec_TAF_LinuxST_vX.X.doc (part 2) to overwrite to the original

3. Download Linux TC script and embed TCs into Linux FUEGO environment

The scope of this document is process 3. It means that below content focus on embedding Lin
To embed Linux TC into FUEGO environment, user has to implement as following 5 steps:

Step 1 Download and Extract Linux TC Scripts

Package name: Linux_FUEGO_TC_Script_YYYYMMDD
Access to Redmine https://socrm.dgn.renesas.com/documents/6822, download and unzip "Li
Linux_FUEGO_TC_Script_YYYYMMDD folder structure is as below



In which:

FUEGO_script	: FUEGO TCs script
FUEGO_config	: FUEGO TCs configuration
Others	: Supporting script for preparing data before exec

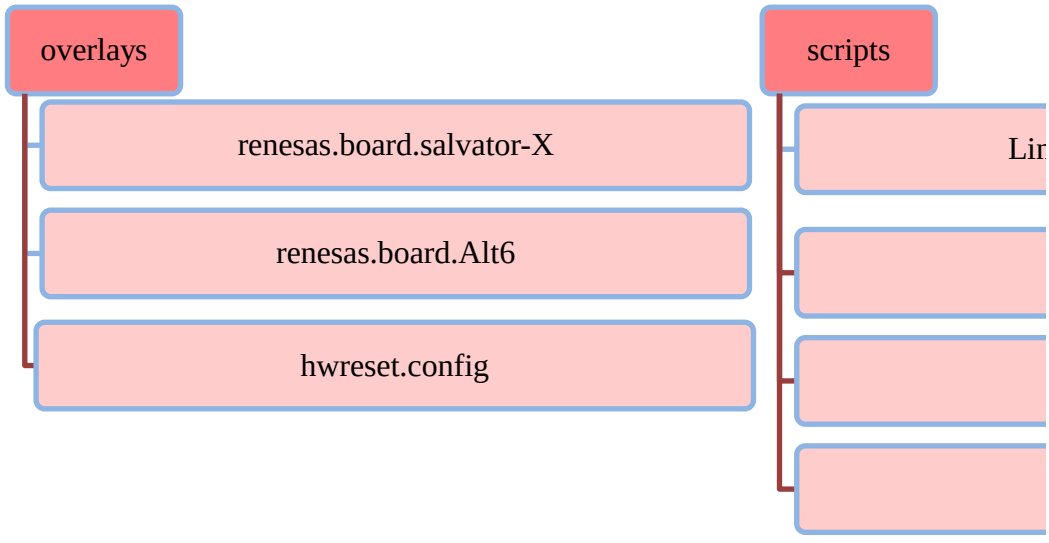
Step 2 **Copy Extracted Linux TC Scripts to FUEGO Directory**

(1) Unzip "Linux_FUEGO_TC_Script_YYYYMMDD.zip" to Linux_FUEGO_TC_Scri

Below is an example of unzip command, in this case, unzipped data is stored at the same fold

```
# cd /home/
# ls -l Linux_FUEGO_TC_Script_YYYYMMDD.zip
-rw-rw-r--  1 rvc  rvc      1550662 Jul 15 18:06 Linux_FUEGO_T
# unzip Linux_FUEGO_TC_Script_YYYYMMDD.zip
```

Linux_FUEGO_TC_Script_YYYYMMDD/FUEGO_script folder structure includes 4 folders



* overlays: added configuration files to support Renesas Board (R-CarH3, R-CarM
* scripts: modified LinuxST_function.sh to support more drivers. In this case, RV
  - For convenience, overwrite original LinuxST_function.sh on FUI
  - Modify LinuxST_function.sh base on the one in the downloaded
  - Add ST_device which contains script may put to device as same a
* tests: includes Linux TC scripts. Each TC corresponds to a folder which contain
  - A host_script.sh to start TC from Linux PC
  - A device folder contains device_script.sh and other script/apps/lib
* r_hwreset_util: includes source and makefile for Hwreset board

Note: In Linux_FUEGO_TC_Script_YYYYMMDD/* package, RVC only release scripts/file

(2) Copy test script into FUEGO folder

Note: REMEMBER to back up replaced files for worst cases

```
# cd /tftpboot/Linux_FUEGO_TC_Script_YYYYMMDD/FUEGO
# cp -R tests/ /home/jenkins/
# cp -R overlays/ /home/jenkins/
# cp -R scripts/ /home/jenkins/
# sudo chmod 777 -R /home/jenkins/tests/
# sudo chmod 777 -R /home/jenkins/overlays/
```

Step 3 Prepare Test Data

Before executing TCs, user need to prepare data depend on execution category

Execution of category TCs, for preparing data before executing TCs

- Application TCs: TCs require special applications which may need to prepare before execut
- Multimedia TCs: TCs require stream data in driver before executing
- Driver TCs: TCs require transfer data in driver before executing

(1) Prepare transfer data for driver TCs

After FUEGO environment setup, and before executing driver TCs, transferring data should b

- File name: bin_xMB_y (x: size of file, y: number of file)
  - E.g: bin_100MB_1 : the 1st binary file (100MB)
  - *** file size, unit of file and file number are named correctly to TC pr**
- Source binary file for copying MUST be put at **OUTGOING** folder of test dev
  - E.g: transfer 5 GB files from NFS to USB, data must be prepared as
    /tftpboot/OUTGOING/bin_1GB_1
    /tftpboot/OUTGOING/bin_1GB_2
    /tftpboot/OUTGOING/bin_1GB_3
    /tftpboot/OUTGOING/bin_1GB_4
    /tftpboot/OUTGOING/bin_1GB_5
- For convenience, copy and execute scripts in Linux_FUEGO_TC_Script_YYY
  + **make5files.sh**

    For transferring large files (5 x 1GB files) TCs, make5file.sh

§ E.g: execute **./make5file.sh /tftpboot/OUTGOING/**
=> create 5 bin 1GB files in folder /tftpboot/OUT
Free space demand: 5GB

+ make5files500MB.sh

For transferring large files (5 x 500MB files) TCs (eMMC t
§ E.g: execute **./make5file500MB.sh /tftpboot/OUTGOI**
=> create 5 bin 500MB files in folder /tftpboot/O
Free space demand: 2.5GB

+ make100files.sh

For transferring many files (100 files) TCs, make100file.sh
§ E.g: execute **./make100file.sh /tftpboot/OUTGOING/**
=> create 100 bin files in folder /tftpboot/OUTG

Free space demand: 5.5GB

(2) Prepare stream data for multimedia TCs

After FUEGO environment setup, and before executing driver TCs, stream data should be pre

- File name: e.g. h264_cbp_l10_128x96_30p_64kbps.3gp
  *** media infromation file are defined correctly to TC procedure in T**
- Source binary file for copying MUST be put at **STREAM** folder of test device
  § E.g: transfer h264_cbp_l10_128x96_30p_64kbps.3gp from NFS to S
  /tftpboot/STREAM/h264_cbp_l10_128x96_30p_64kbps.3gp

(3) Prepare application for application TCs

Before executing application TCs, below applications must be prepared, please refer to Appe

App name	Gen3 (Guideline's link)
Benchmark_apps	Benchmark
Busload	Busmoni
CodecTP	CODECTP
image_libs	image_libs
OpenGL	OpenGL
ts_calibrate	ts_calibrate
ttyecho	ttyecho
SCIF	SCIF
viewport-scaler	Lack of application.
OES2_Texture_Pixmap_EGLImage_wl	Lack of application
Busybox	Refer to Implementation_Spec_TAF_LinuxST_v1.3.doc for installation (part 1.4.2)

For more detail information about test application, please refer to Appendix sheet

Note: Total amount of data is very huge, so depend on capacity of device. With manual or se
With full automation TCs, the TCs should separate into groups for preparing data effectively.

Step 4 Embed TCs to FUEGO Environment via FUEGO Web Interface

After step 2, TCs and configuraition files are copied into FUEGO directory only. This action
A TC can only be executed on a target board when that TC and the configuration file of that b

0. Open FUEGO web interface

Start Docker by command line on Host PC:

/opt/fuego/fuego-host-scripts/docker-start-container.sh

Launch Firefox and access link: localhost:8080/fuego to launch FUEGO web interface

1. Add new node (board/device)

(This step MUST be omitted if target board/device are available in Target Status tab of Home

Each board/device has its corresponding configuation file in overlay folder. To realise this co

(1) Select Target Status

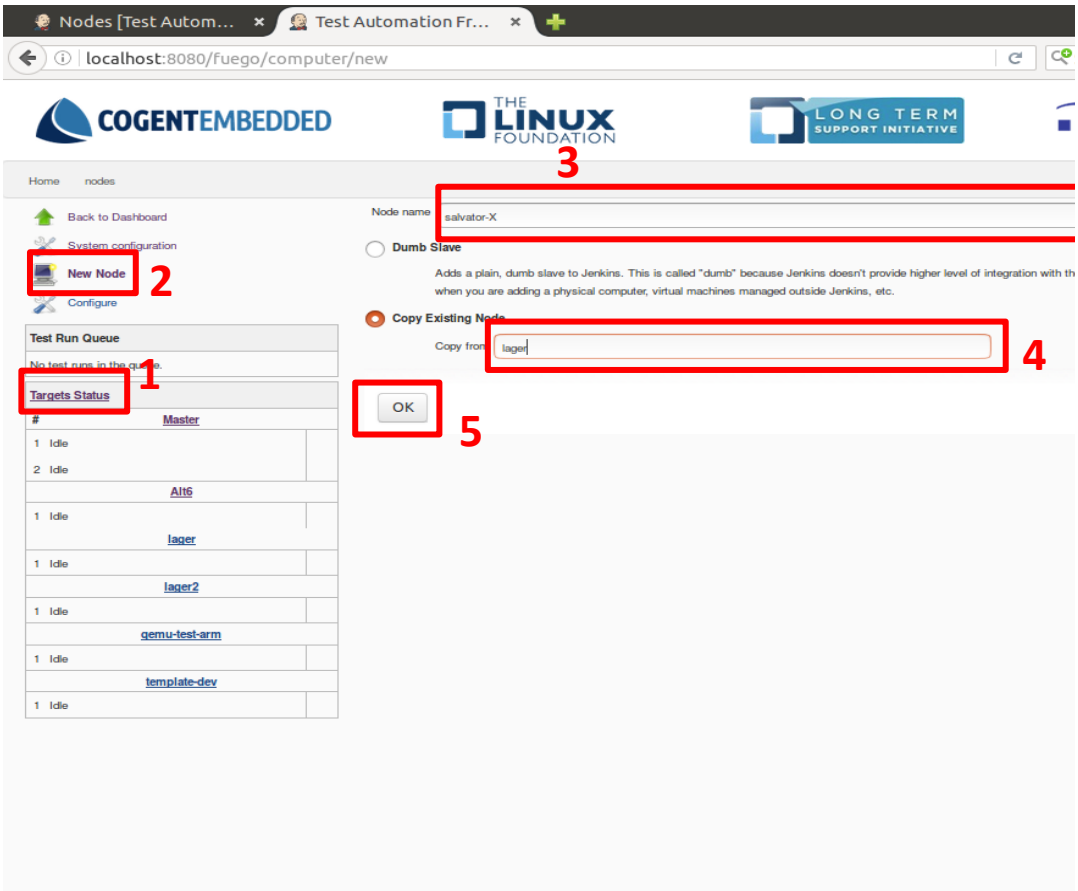
(2) Select New node

(3) Name of the node which be newly added (e.g., salvator-X)

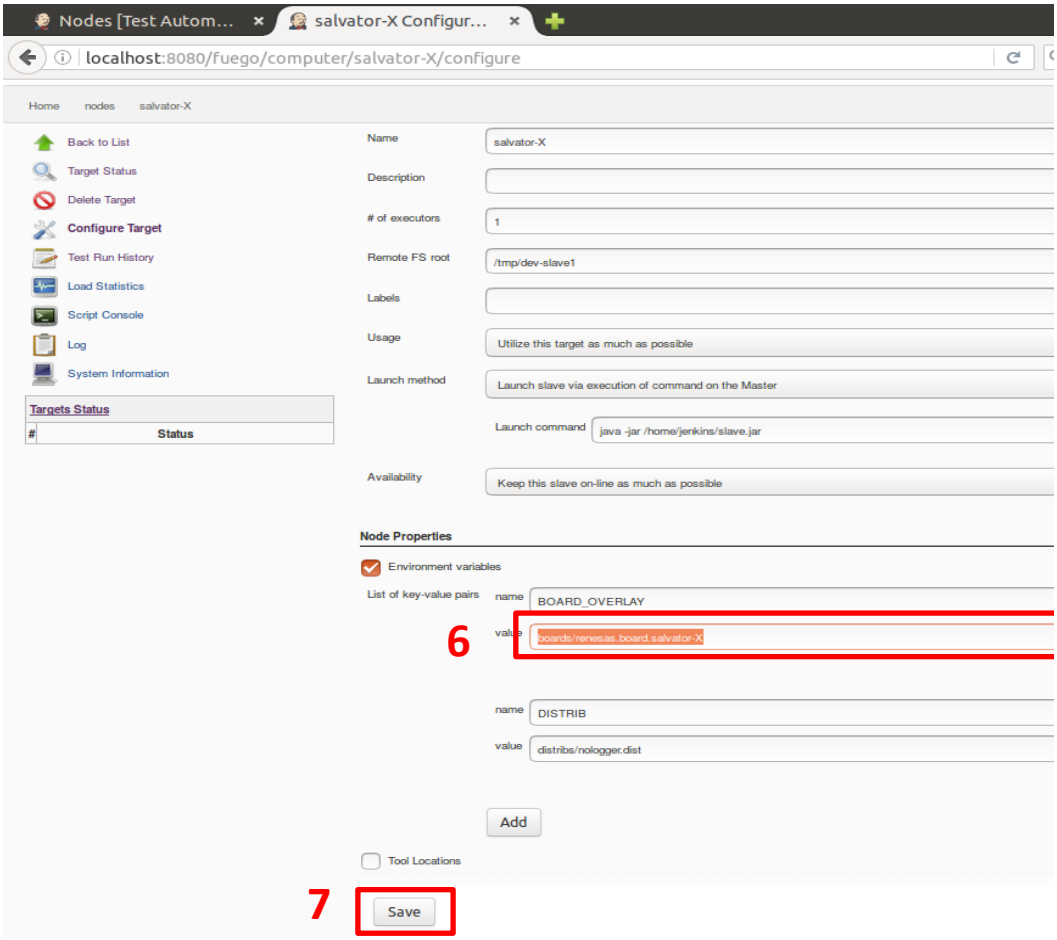
(4) Copy available settings of existing node (e.g., lager)

* lager is an available node that is implemented as a standard device in FUEGO e

(5) Select OK

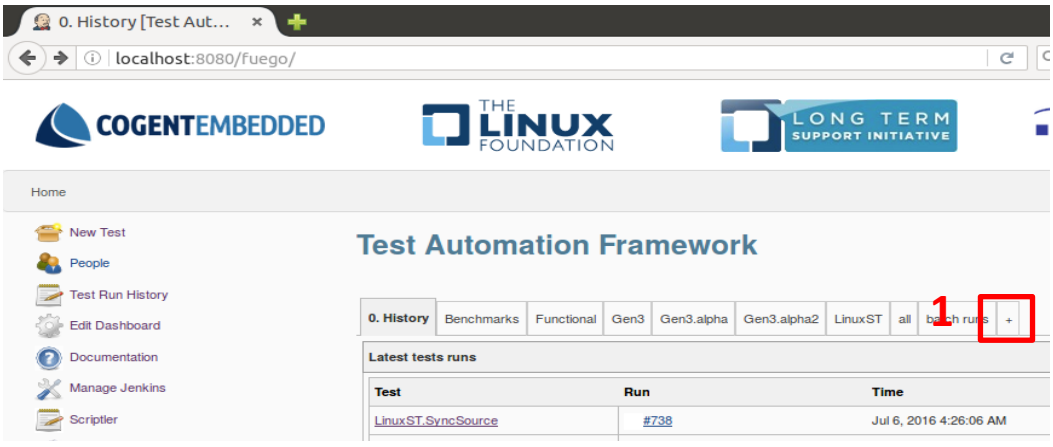


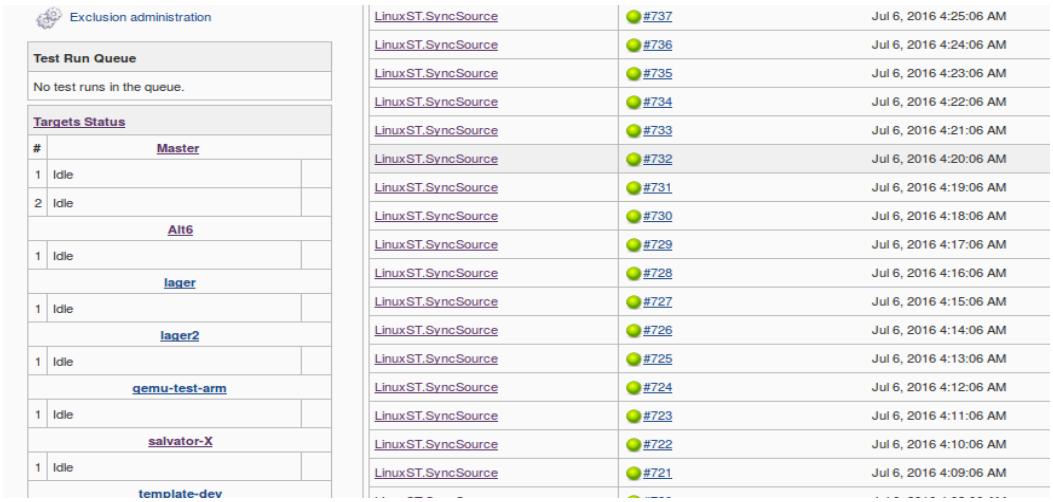
(6) Directory of corresponding configuration file in overlay folder (e.g., board/rene
(7) Select Save to save all settings



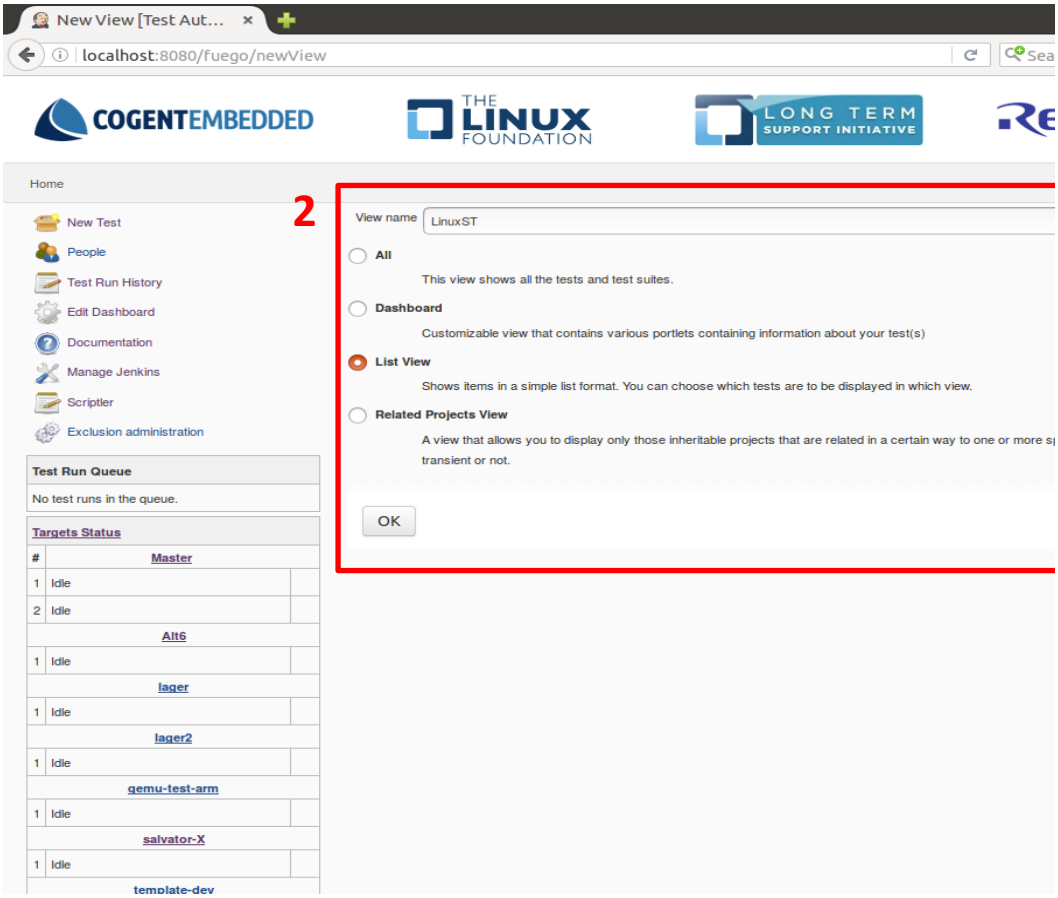
2. Add new tab (to manage TCs)
Due to a large number of TCs, they are divided into different groups (all, batch run, Benchma
For example, to group all Linux TCs together, RVC created a new tab called Linux ST. Pleas

(1) In Home page of TAF, click to "+" tab to add new tab

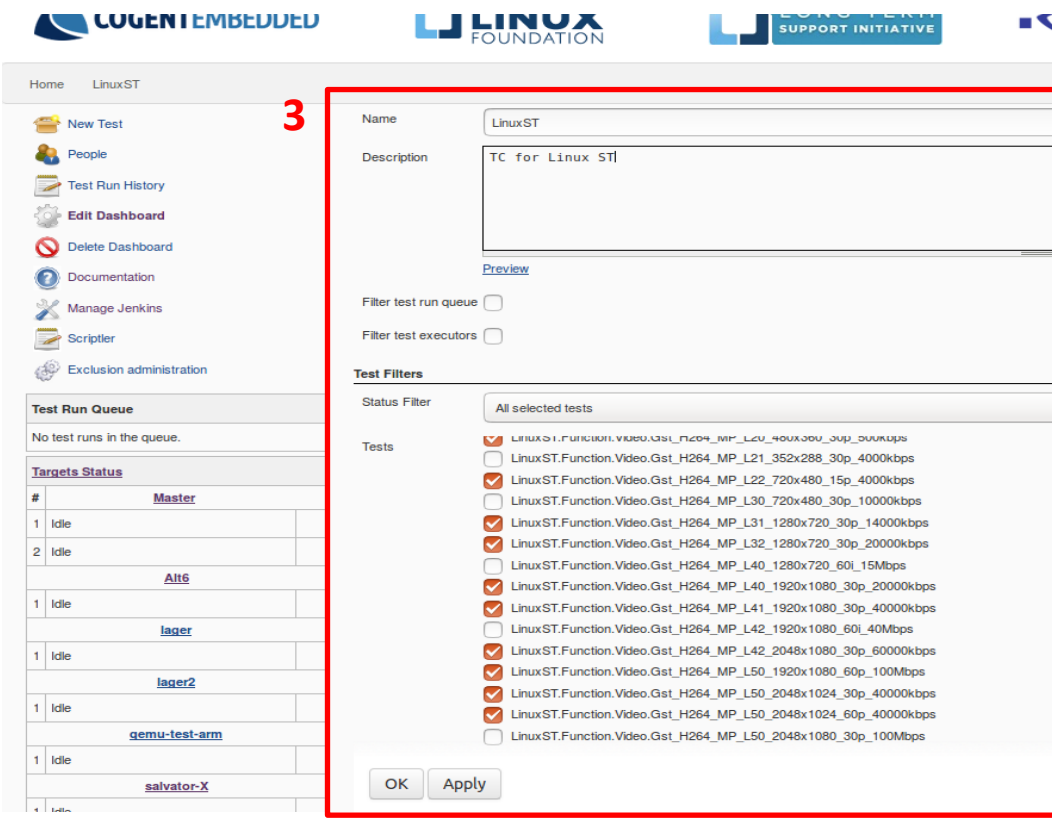




(2) Name for tab (e.g. Linux ST) and select view type of tab:
(List View is recommend for simple list format of choice tests)



(3) Add description for tab and select available TCs that user want to include in ta

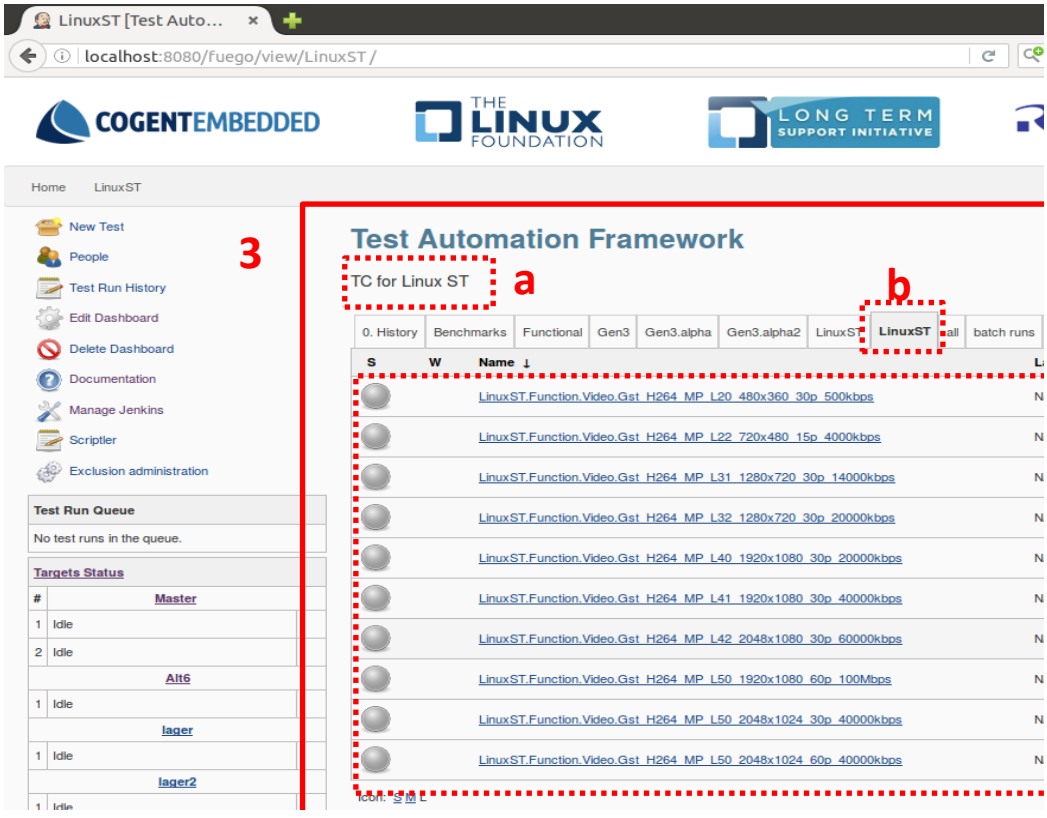


(4) Interface of Linux ST tab which was created

a. Tab description

b. Tab name

c. List of TCs in tab



3. Add an Linux TC to FUEGO environment

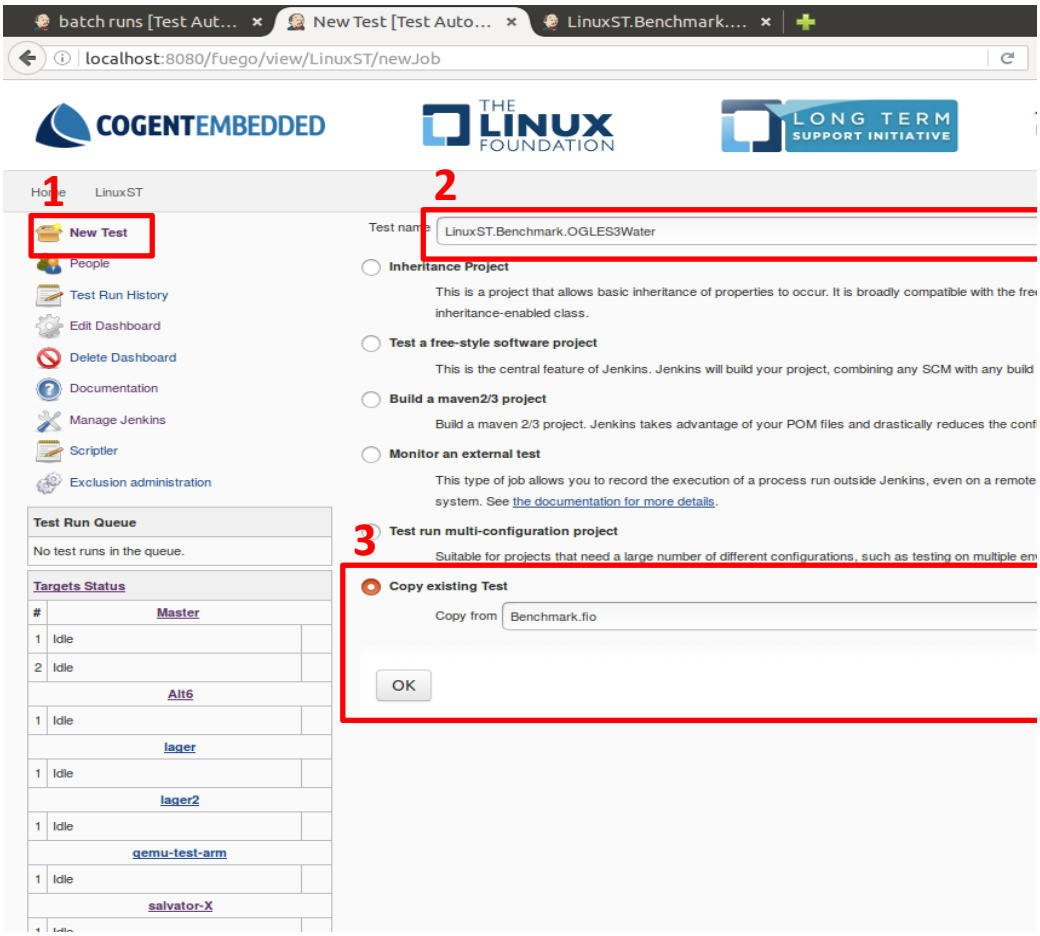
To embed the scripts of an Linux TC to FUEGO environment, do step-by-step as below pictu

(1) In web interface, select New Test option to create a new TC

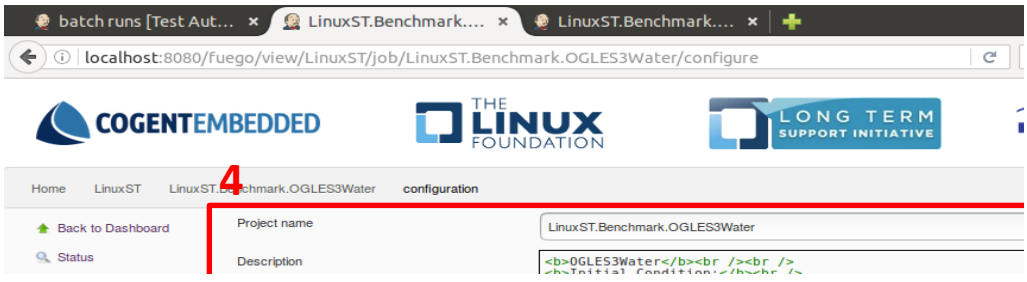
(2) Fill out TC name in Test name blank, this name must be same as the folder nai

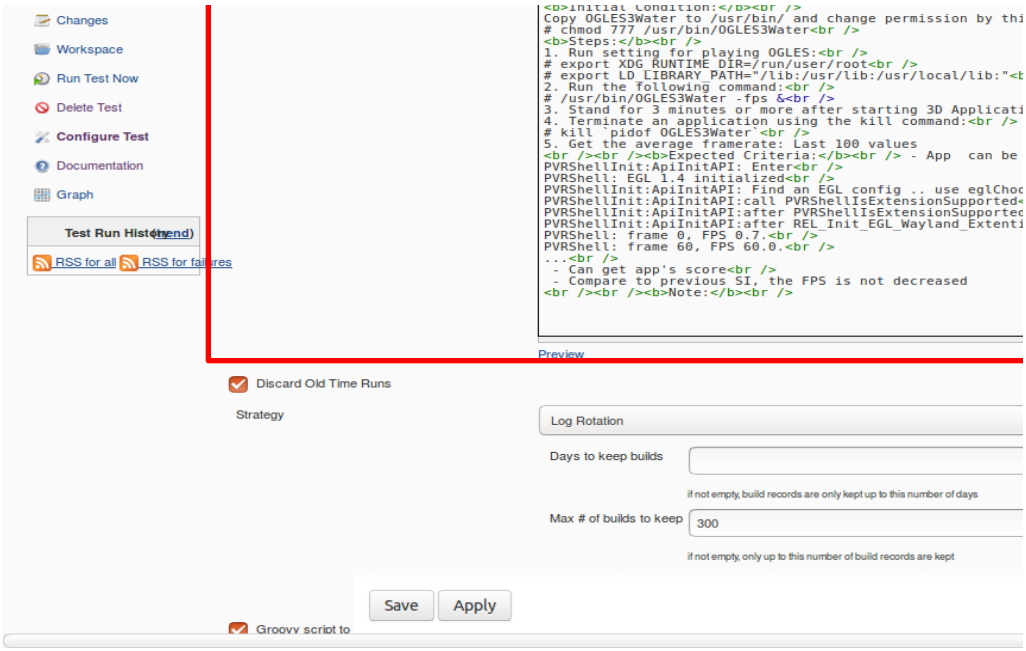
(3) Copy available settings of existing test (e.g., Benchmark.bc)

* Benchmark.bc is an available test that is implemented in original FUEGO envirc



(4) Check TC name and Insert TC description into blank





(5) (6) (7) (8) (9) are the options that appear on the test execution interface (3rd re

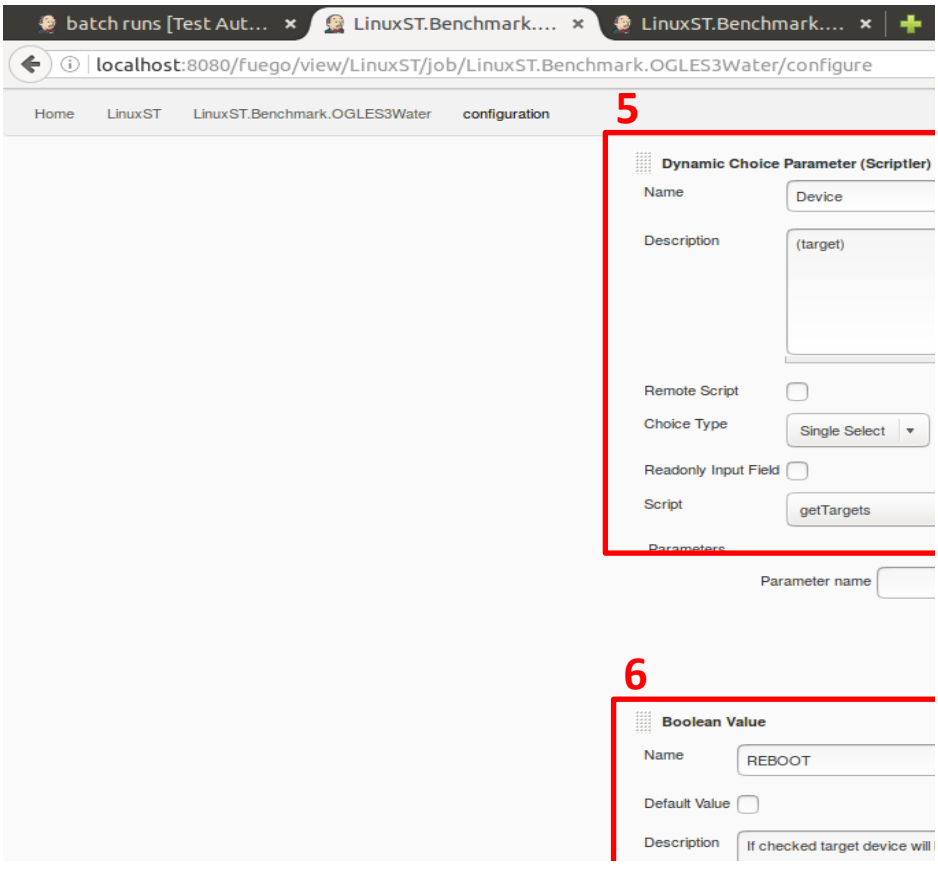
* Upper case and lower case are distinguished

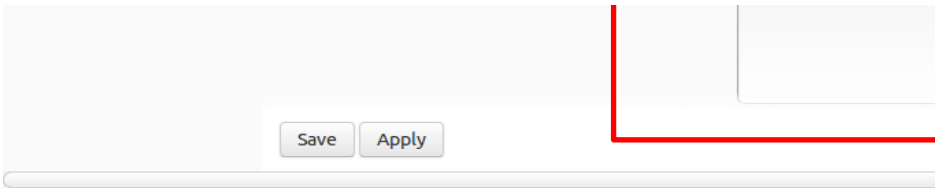
(5) Device - single select box to select the target device (e.g. koelsch)

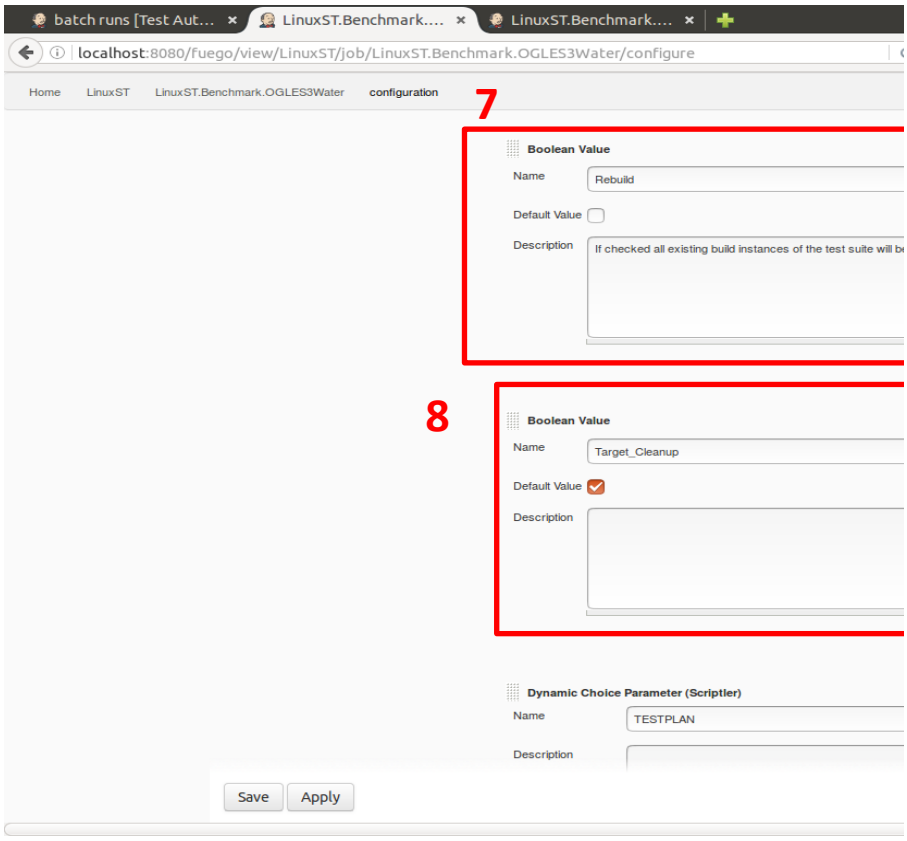
(6) Reboot - check box to reboot target device before running test

(7) Rebuild - check box to rebuild test suite from tarball (device_script.

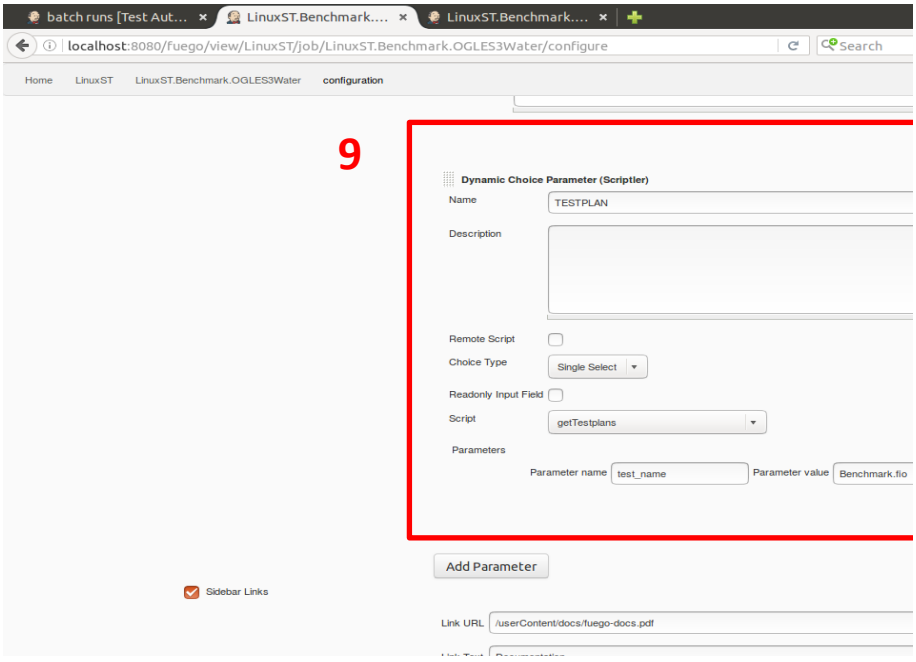
(8) Target_Cleanup - check box to clean all old log file in target board

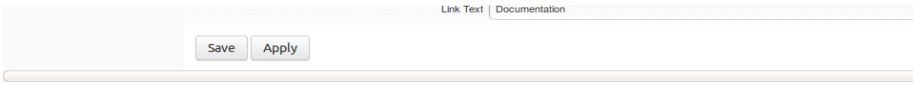




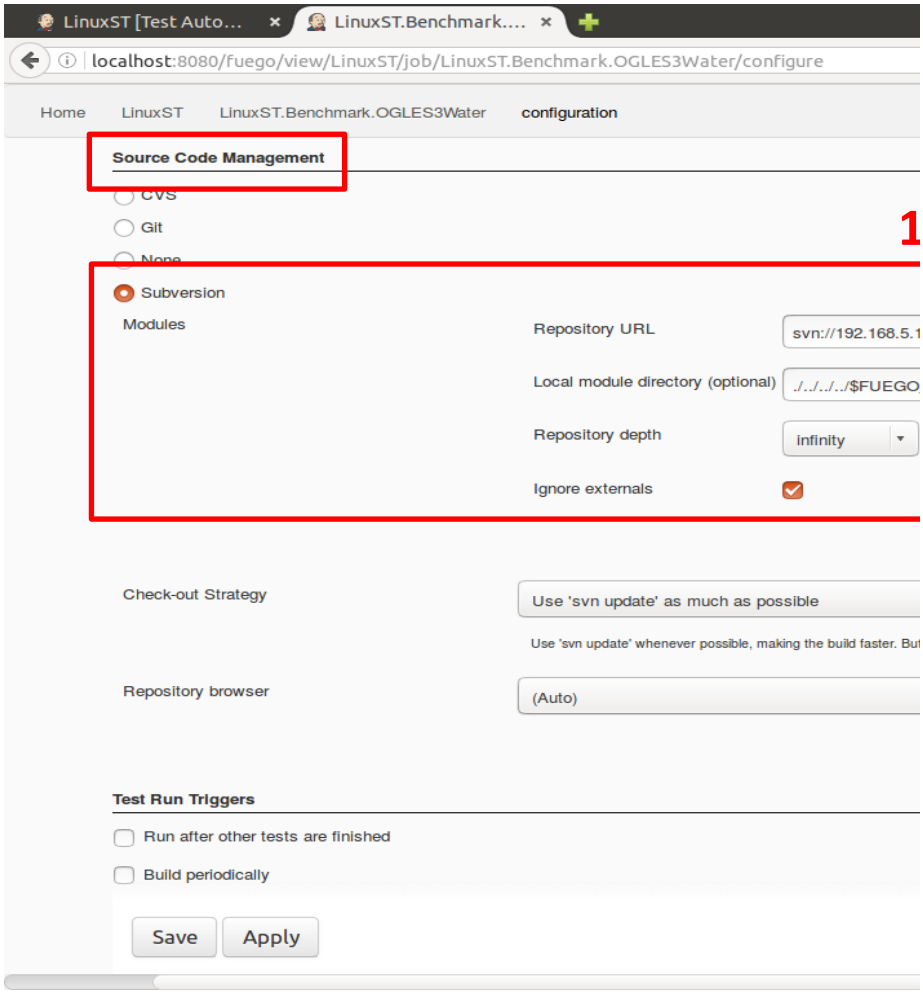


(9) Test plan: Delete this field, currently not used.

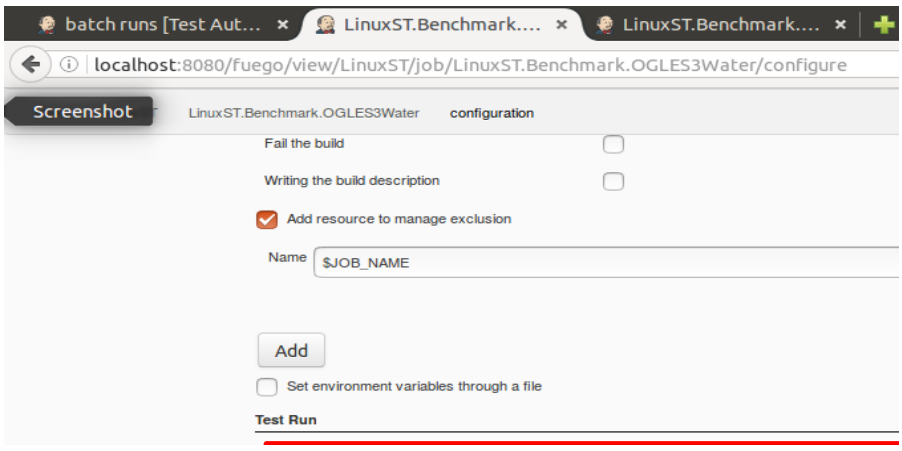


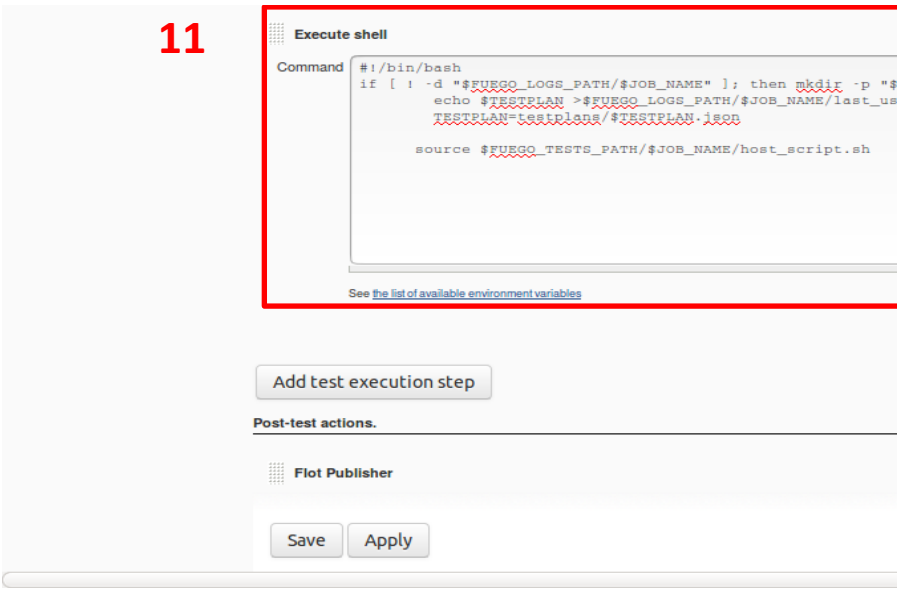


(10) Enter Repository URL and Local directory

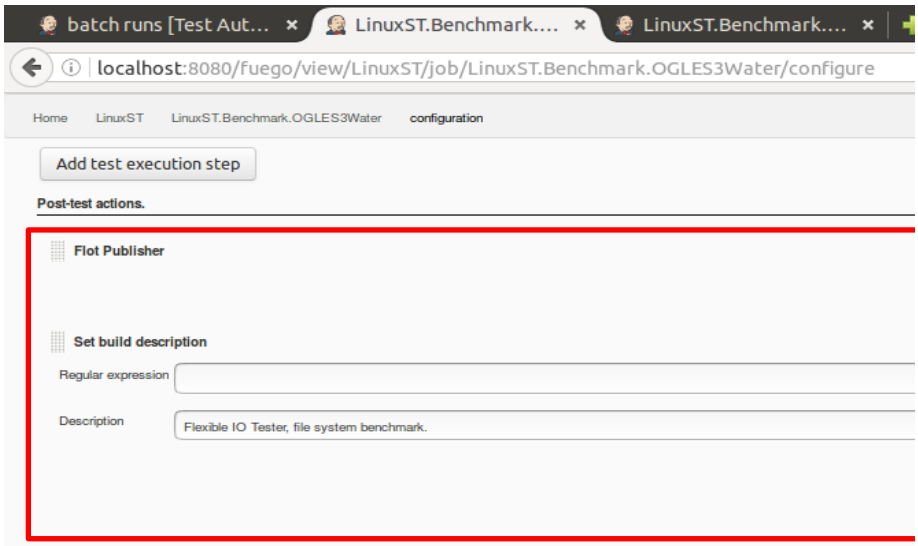


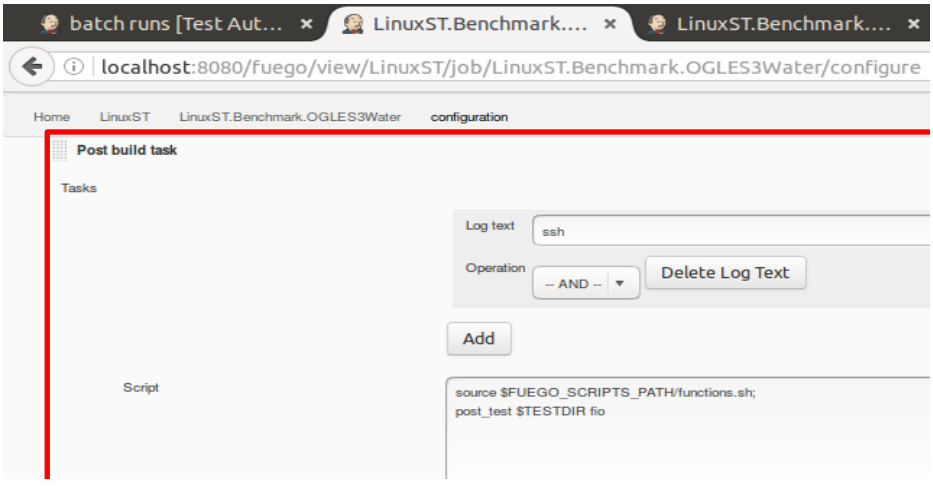
(11) Test Run - adjust to execute test (host_script.sh)
"#!/bin/bash" need to be added into the beginning of Execute shell if it

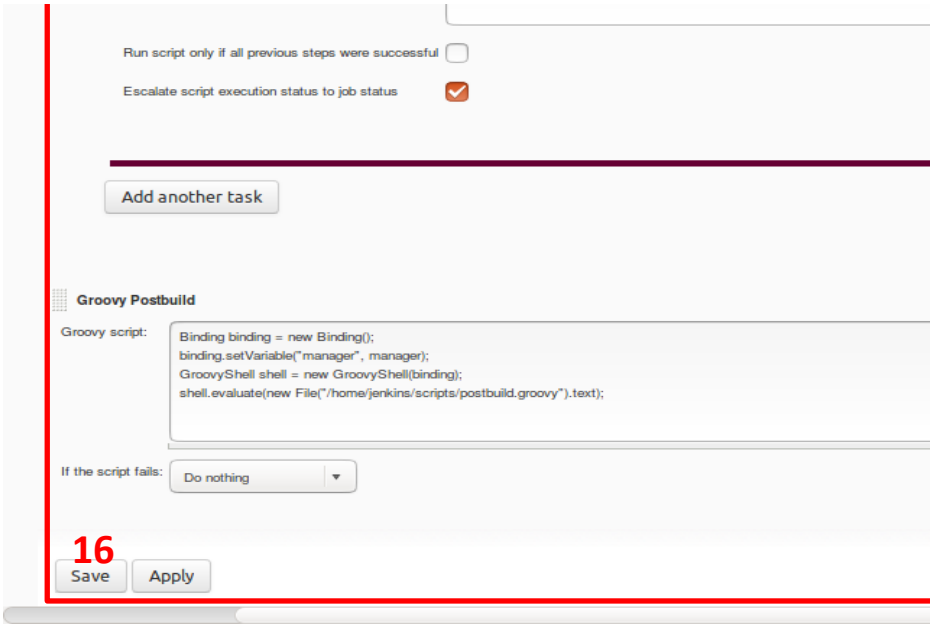




(12)(13)(14)(15) Post-tests actions - delete these actions (currently not s
(16) Save changes of test case.







Step 5 Execute TC and get the result

NOTE:

- At first, should run test case **"LinuxST.First_TC.Install_Applications"** to install all of ne

=> Not need to run this Test Case if:

+ All of applications/libraries have already been installed on target boa
+ The User just want to install manually for some specific apps/libs

1. Execute 1 TC

Select a TC and Target device to execute test:

(1) Select TC to be executed

(2) Select Run test now

(3) Select target board and options for TC (all options should be selected)

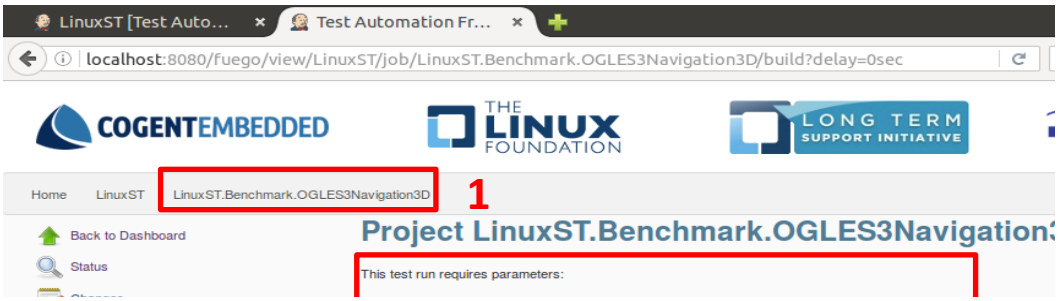
_Device (mandatory): select the target device (e.g. koelsch)

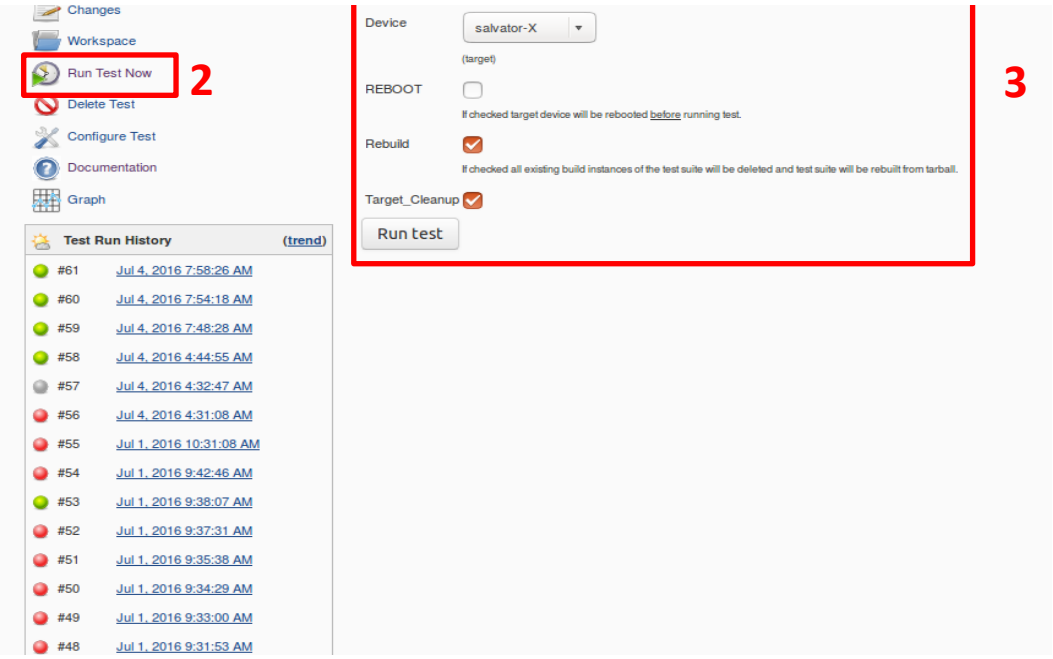
_REBOOT (recommendation) reboot target device before running test

_Rebuild (recommendation): rebuild test suite from tarball. If test suite

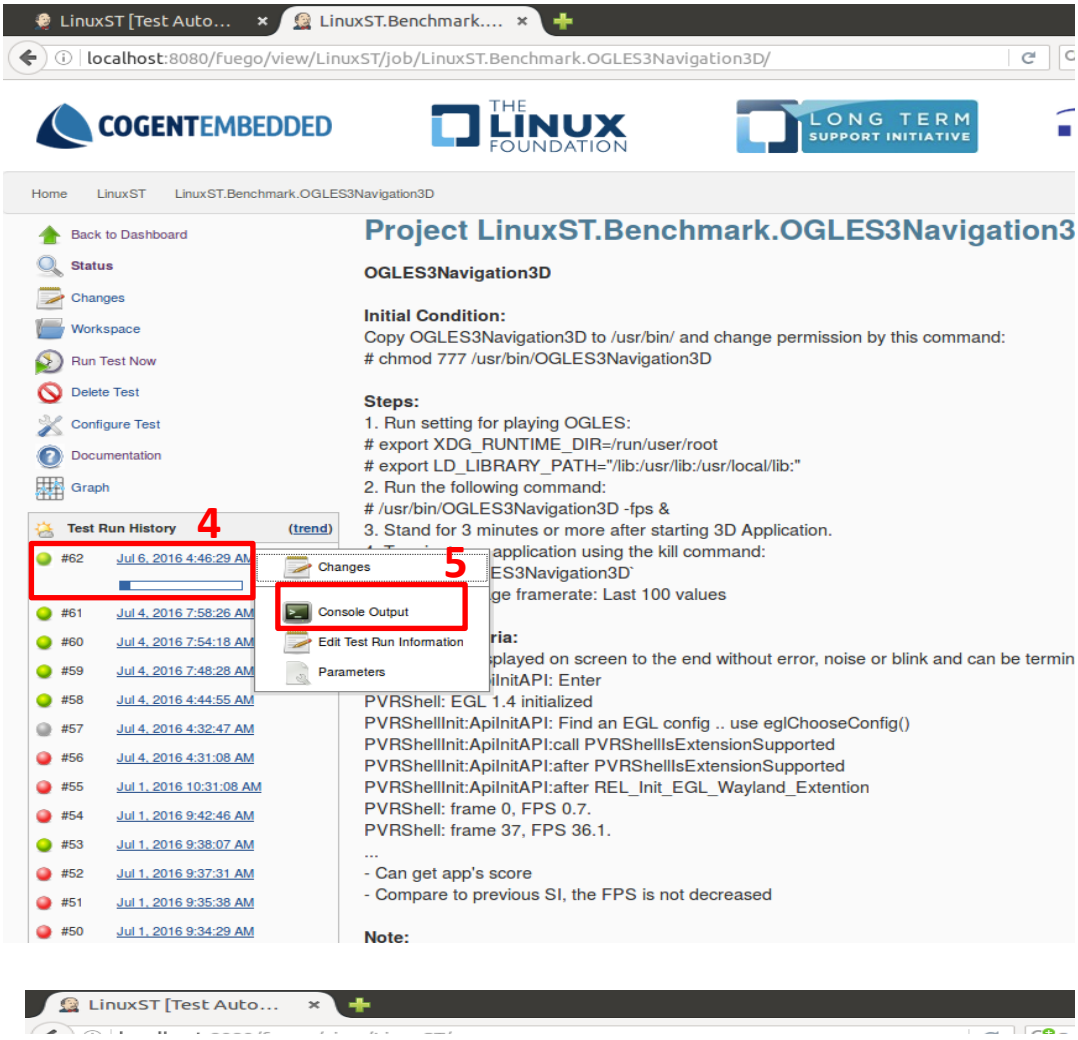
_Target_Cleanup (mandatory): clean all old log file in target board

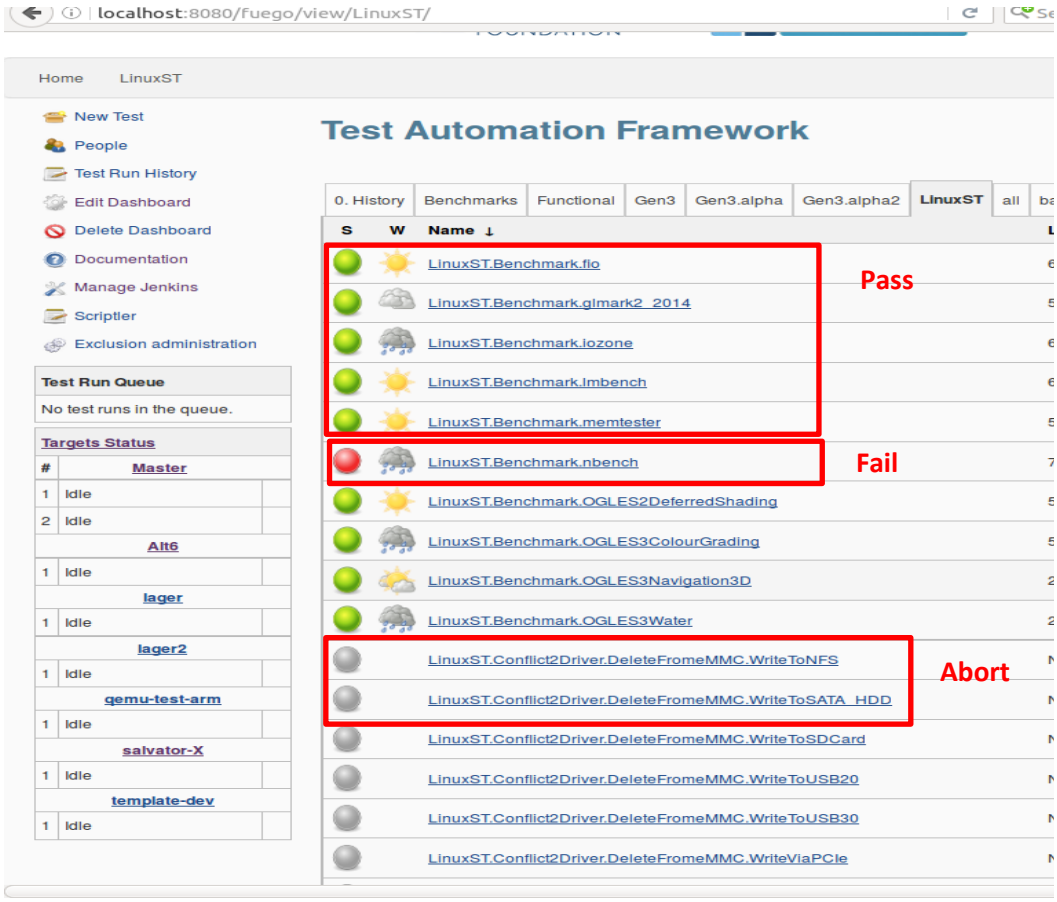
_Run Test: execute test (run host_script.sh)





(4) (5) Show log of executed TC

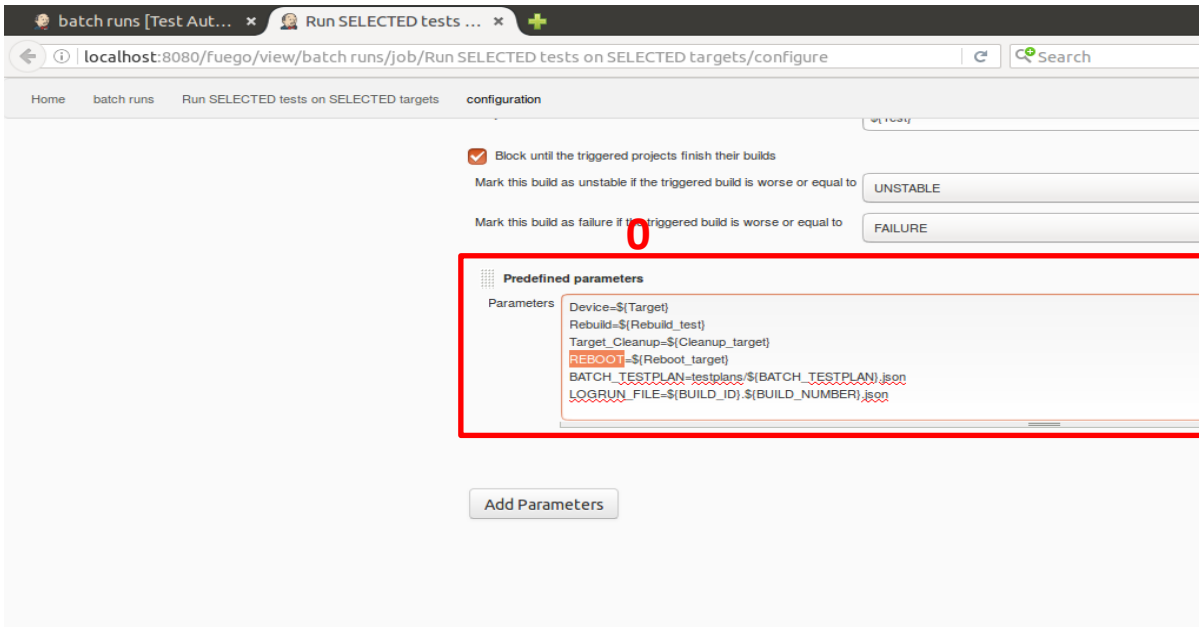


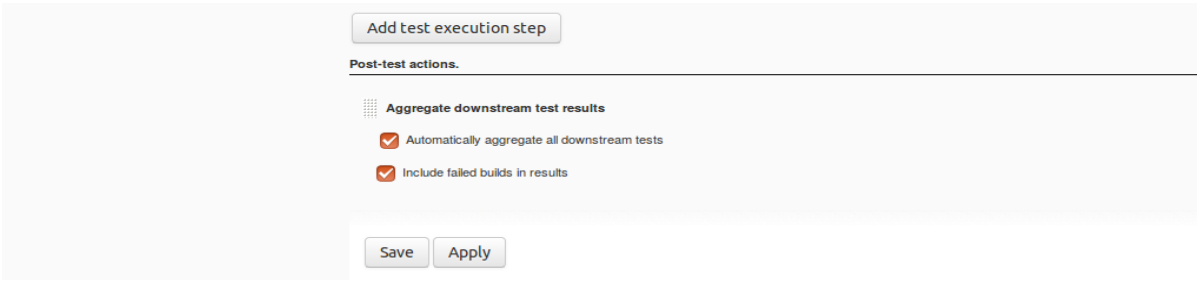


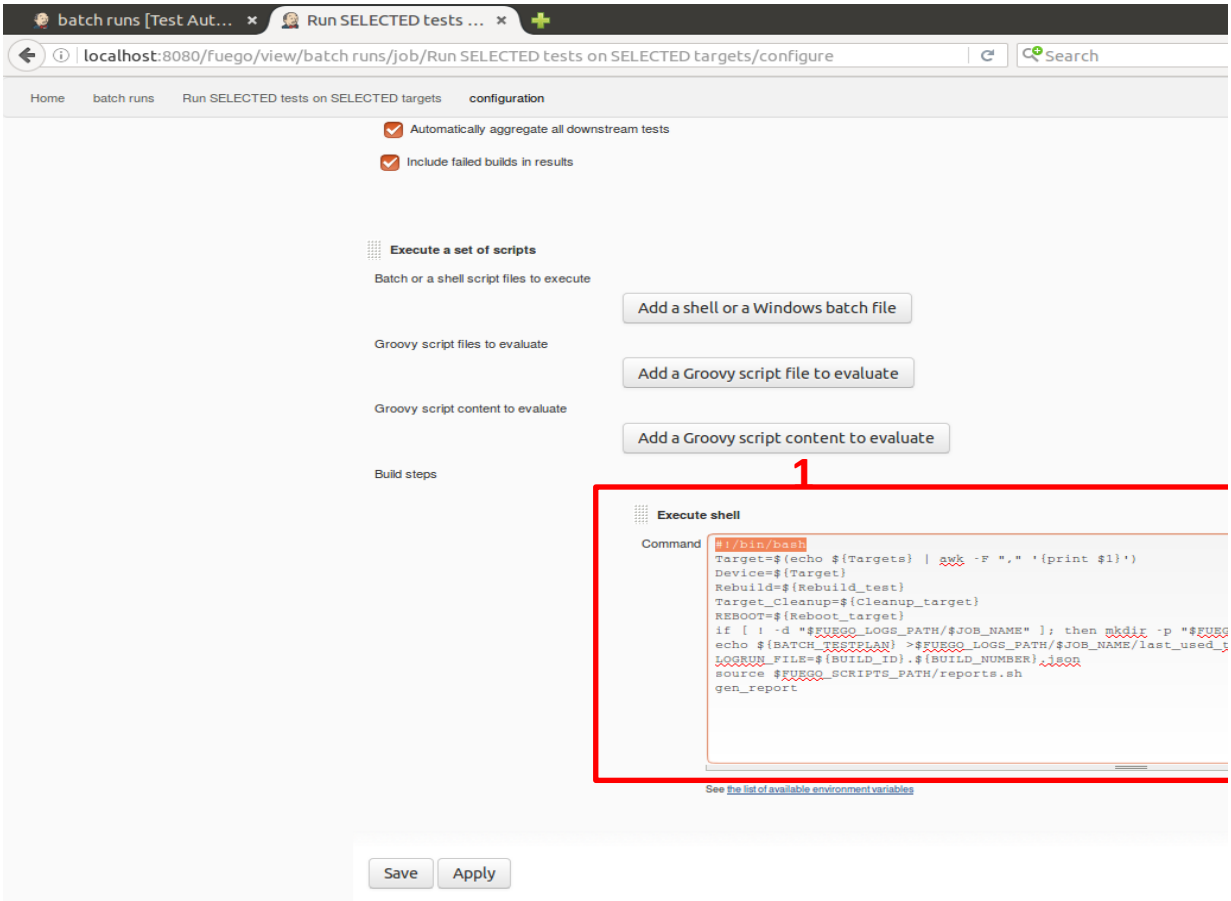
2. Execute many TCs

Run available TC "Run SELECTED tests on SELECTED target" in "batch runs" tab to execu
This special TC allows to run many selected TCs, it means that users have to select TCs that t
Before running, reconfig TC as below:

(0) Changing Reboot parameter into REBOOT to support reboot function when st
(1) Adding #!/bin/bash into Execute shell to support bash shell







Running test procedure:

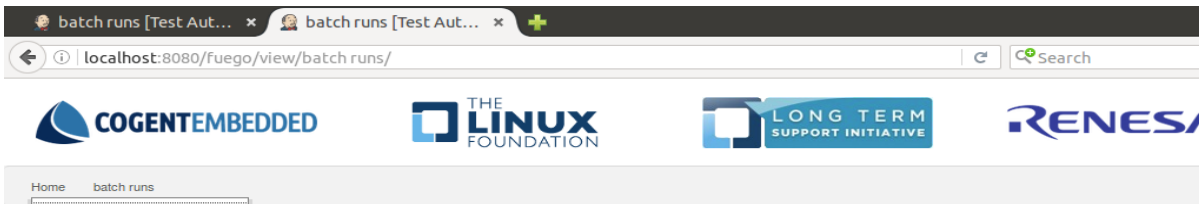
(2) Select batch runs

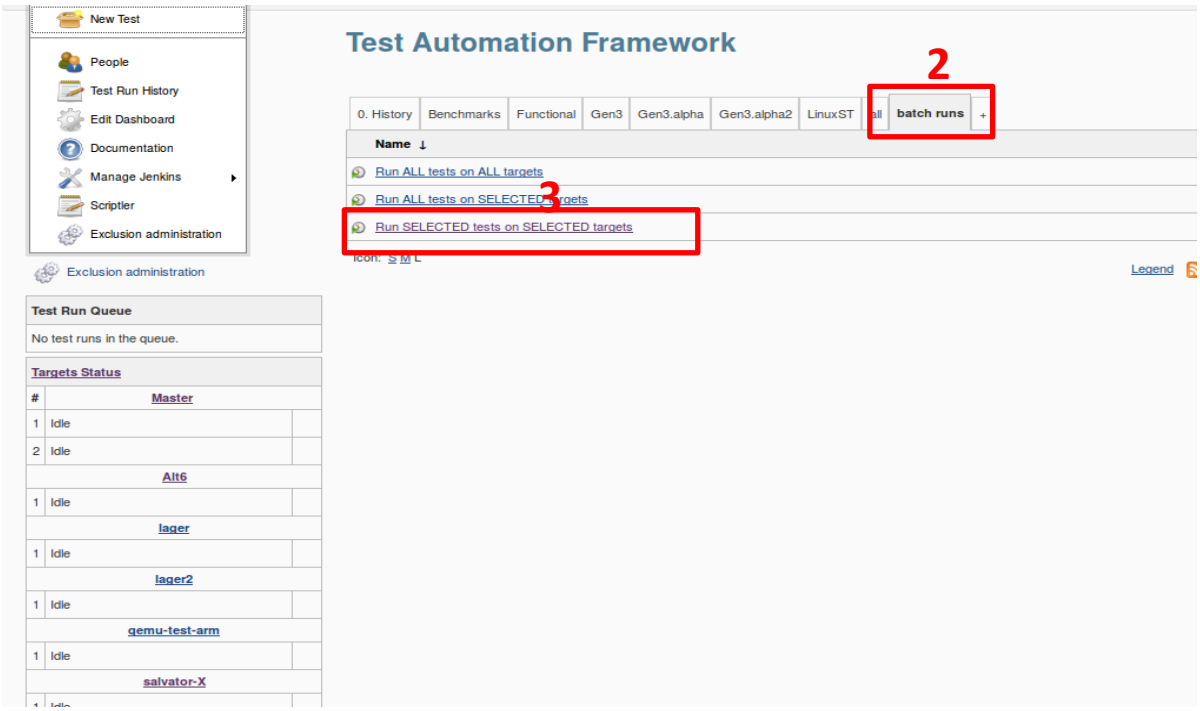
(3) Select TC "Run SELECTED tests on SELECTED target"

(4) Select Run Test Now to execute test

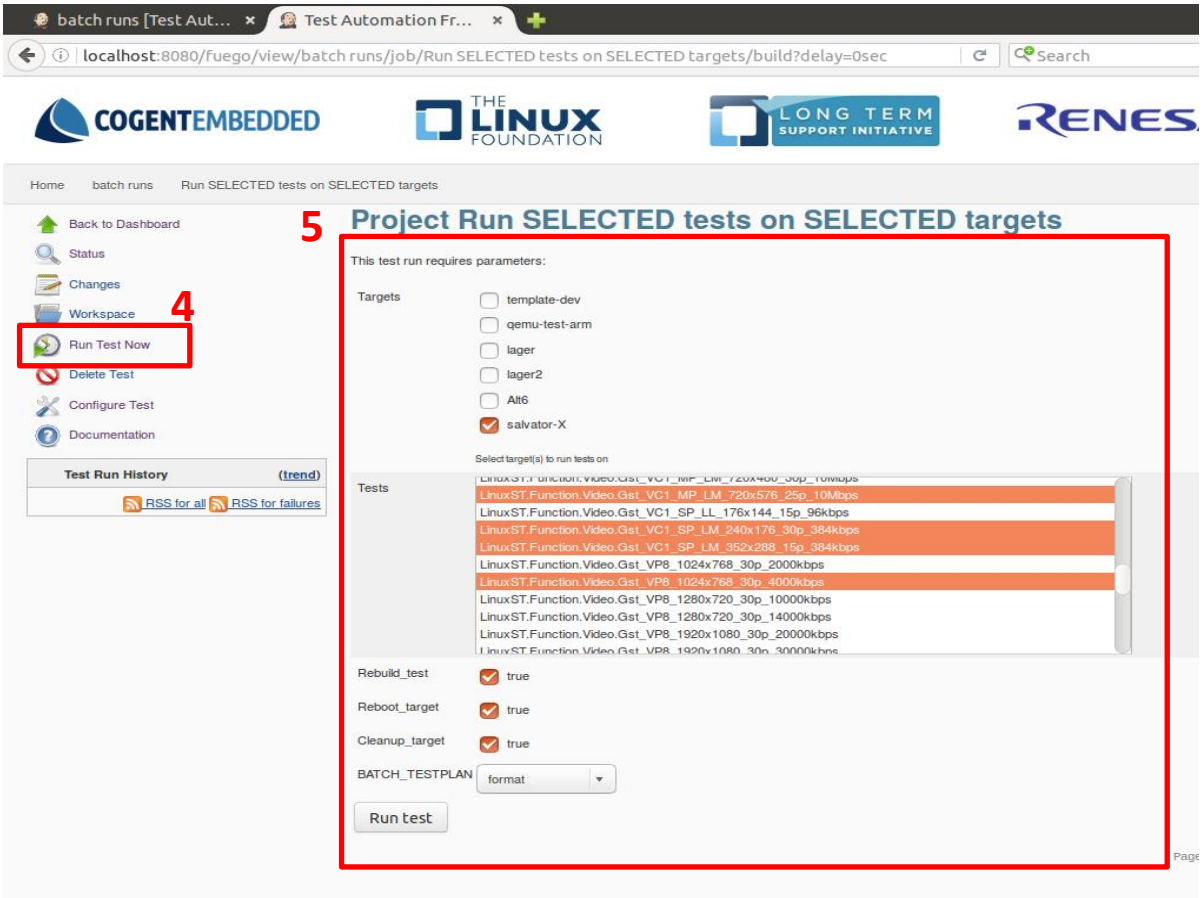
(5) Select target board, TCs that users want to run, other properties (rebuild, reboo

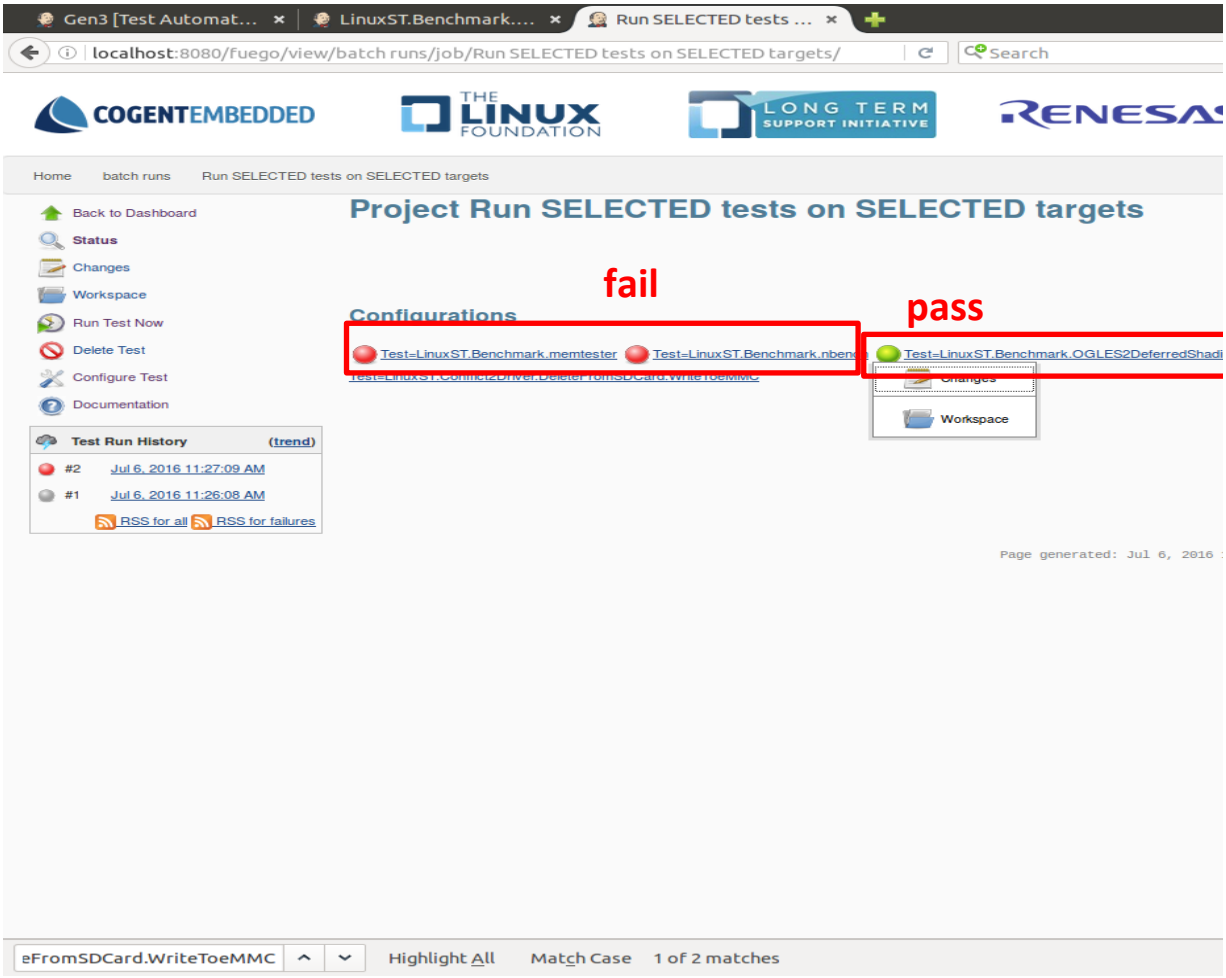
(6) Value test results





Running test:



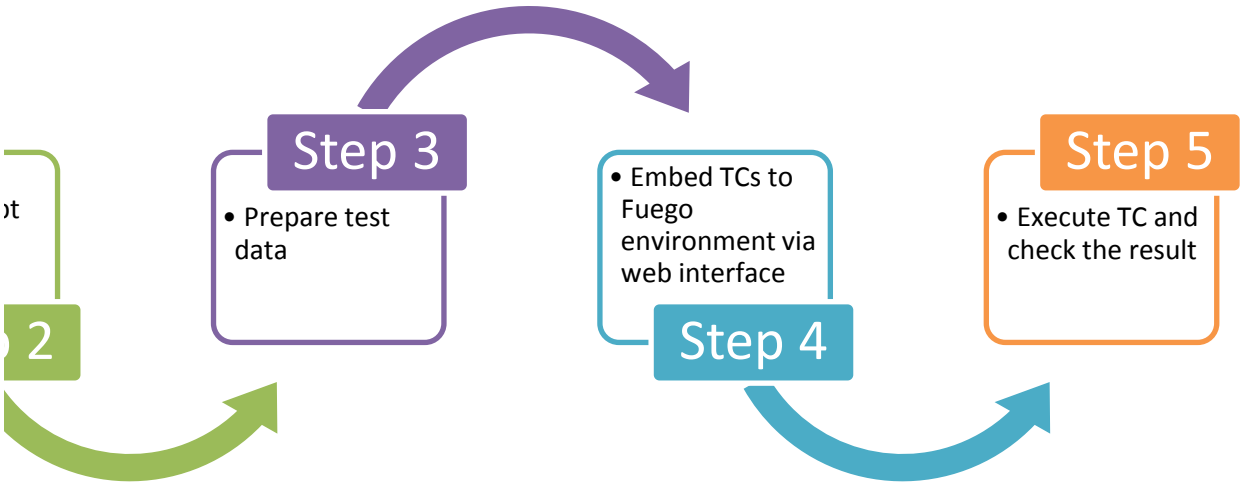


Restriction:

Now, preparing data has to do manually and take a lot of time, RVC will improve by making

MANUALLY

ernal



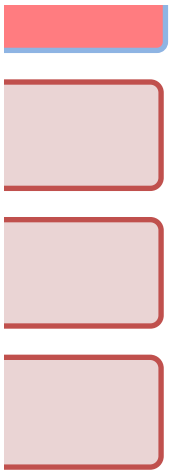
ment first. Three main processes are listed as below:

d/fuego/

FUEGO environment

ıux TC scripts into Linux FUEGO environment only

inux_FUEGO_TC_Script_YYYYMMDD.zip ” (assumption that file is downloaded to /home/ directory)



uting TCs

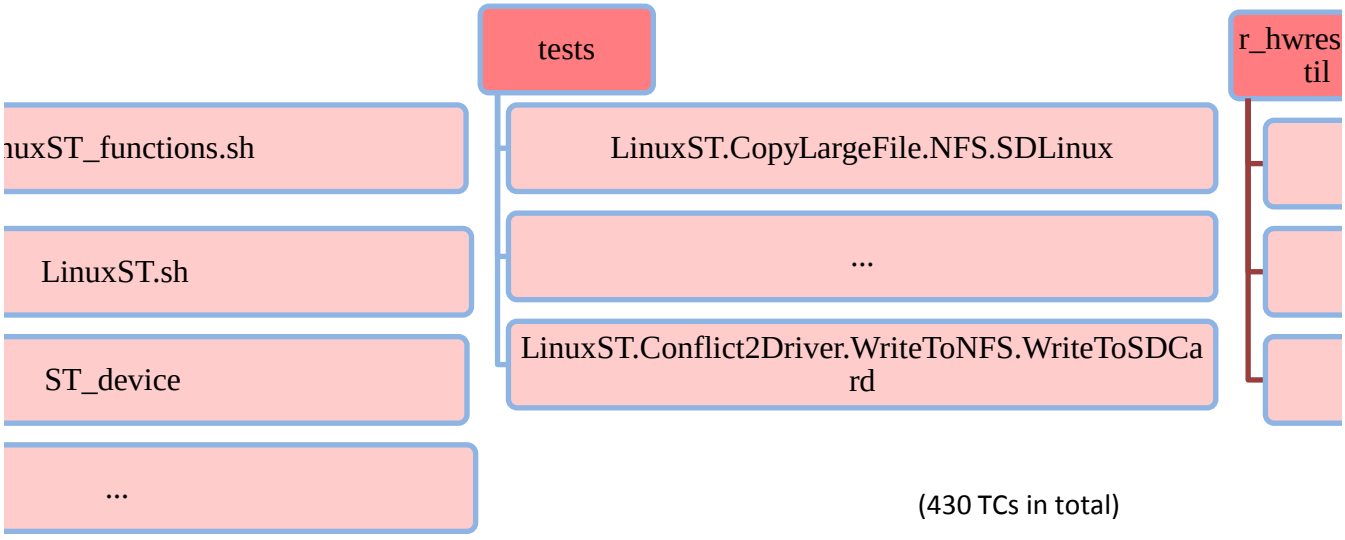
---

pt_YYYYMMDD

ler with zipped data (/tftpboot/):

```
C_Script_YYYYMMDD.zip
```

s: overlays, scripts, tests and r_hwreset_util



M3, R-CarE2X)
C recommend 2 approachs:
EGO env. by the one in downloaded package (Linux_FUEGO_TC_Script_YYYYMMDD/scripts/LinuxS
package
as device_script.sh
s:

brary if necessary to execute TC on target device

s which are modified or created by RVC. Other remined FUEGO files are not included in the package.

```
O_script
```

---

ing

be prepared for each device:

rocedure in TCs_summary sheet. Detail information can be found in Linux_SystemTest_Spec_Gen
vice
below on NFS:

YYMMDD/Others/Data_Creation_Scripts/ on Host PC to create transferred files easily

h script is used to create transferred data

TGOING/

o eMMC), make5file500MB.sh script is used to create transferred data
NG/
UTGOING/

script is used to create transferred data

OING/ which have + file size from 10MB to 100MB
+ 10 files for each file size
+ jump step is 10MB

epared for each device:

TCs_summary sheet. Detail information can be found in Linux_SystemTest_Spec_Gen3_ver3.x.xls

SDCard data must be prepared as below on NFS:

ndix sheet for prepare applications

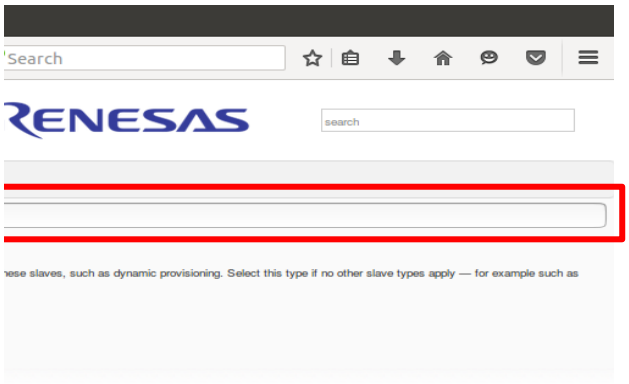
	E2X (Guideline's link)
	Same as Gen3
	Axistat
	Same as Gen3
	Same as Gen3
	Same as Gen3
	Same as Gen3
	Same as Gen3
	Same as Gen3
	Lack of application.
	Lack of application
build and	Refer to Implementation_Spec_TAF_LinuxST_v1.3.doc for build and installation (part 1.4.2)

mi-automation TCs, data could be copied each time before executing.
Please check Pre-execution_Requirement for more information about prepare data before executing

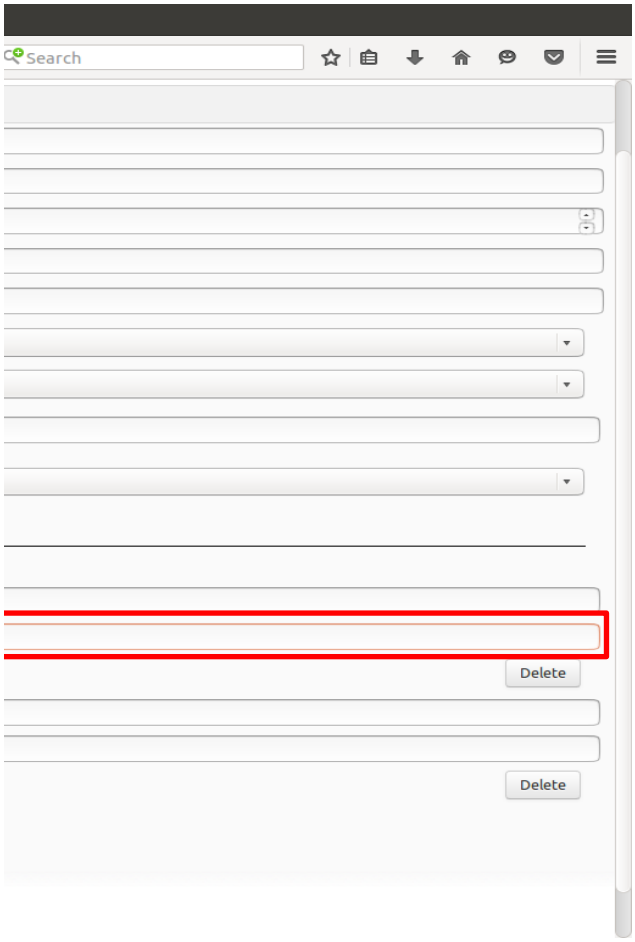
means nothing if they are not embedded to FUEGO environment via web interface.
oard are embedded to FUEGO environment as below 3 steps (please focus on red zone):

interface)
nfiguration file, add it as a new node in FUEGO web interface:

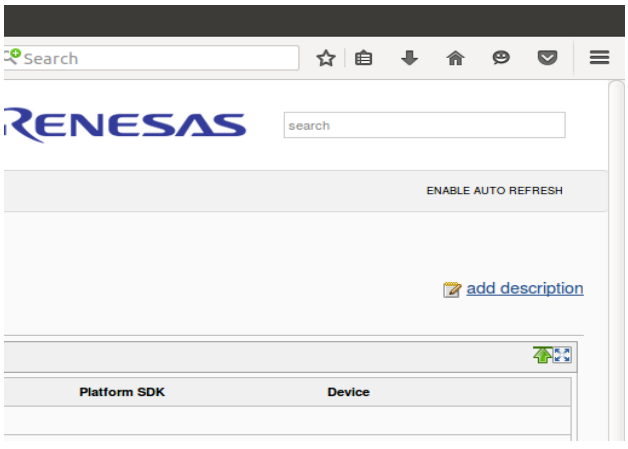
nvironment (Jul 31, 2014)

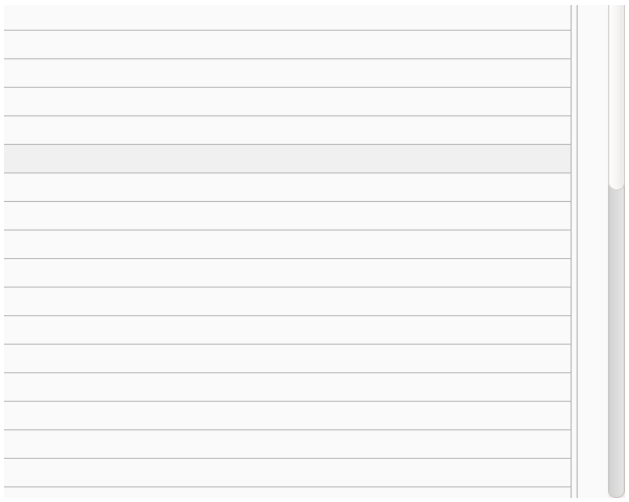


esas.board.salvator-X)



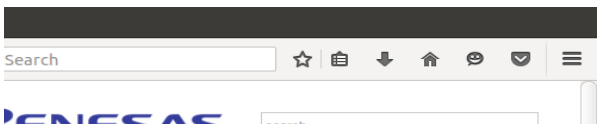
rks, Android ST, Linux ST,…) to manage. Tab concept is the name of group
e refer to below guideline to create the Linux ST tab:

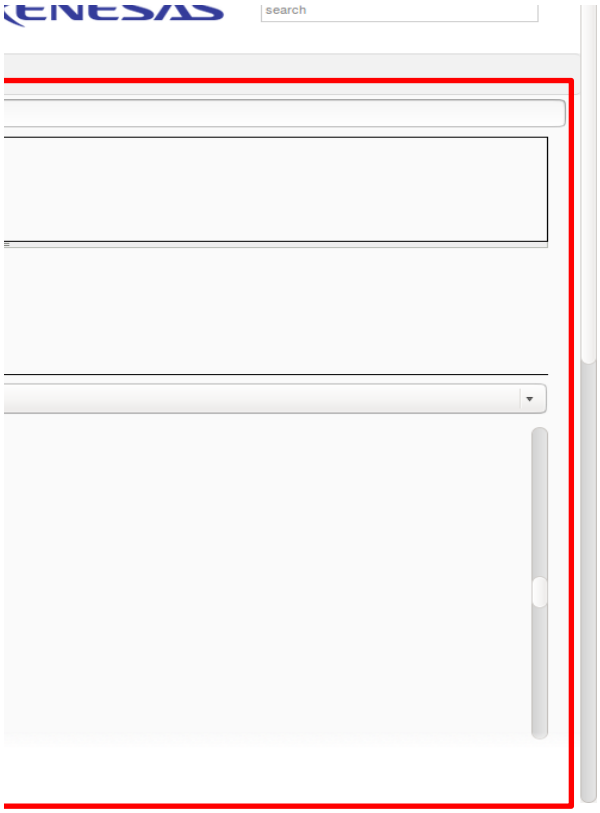


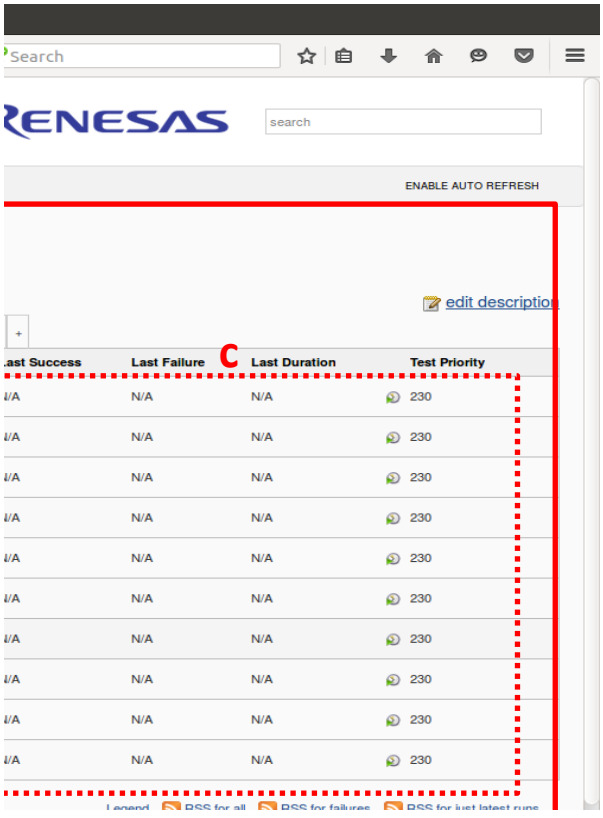



search
pecific projects. Also allows you to filter based on whether or not a particular project is

b


Search

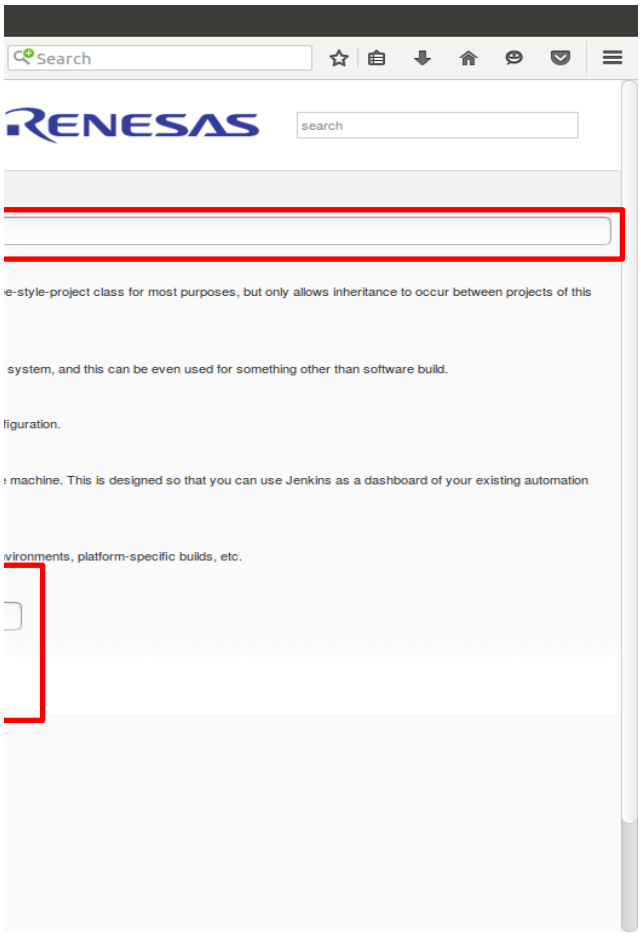

search

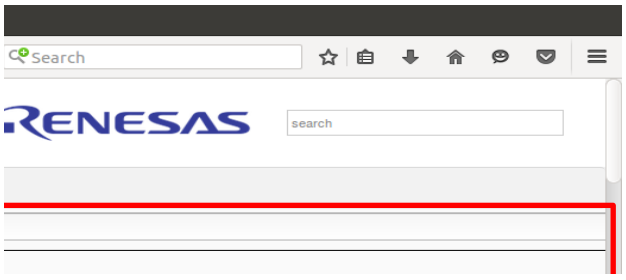

Search
search
ENABLE AUTO REFRESH
edit description
Last Success
Last Failure
Last Duration
Test Priority
N/A
230
RSS for all
RSS for failures
RSS for just latest runs

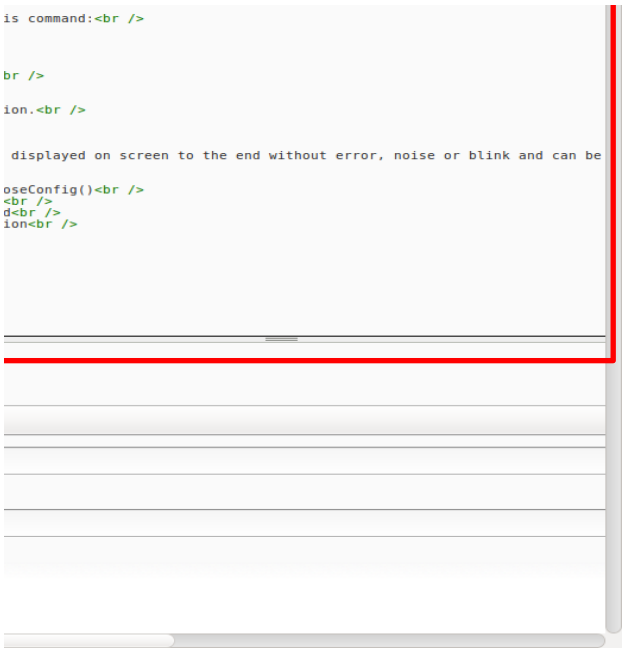
.res:

me of TC scripts

onment

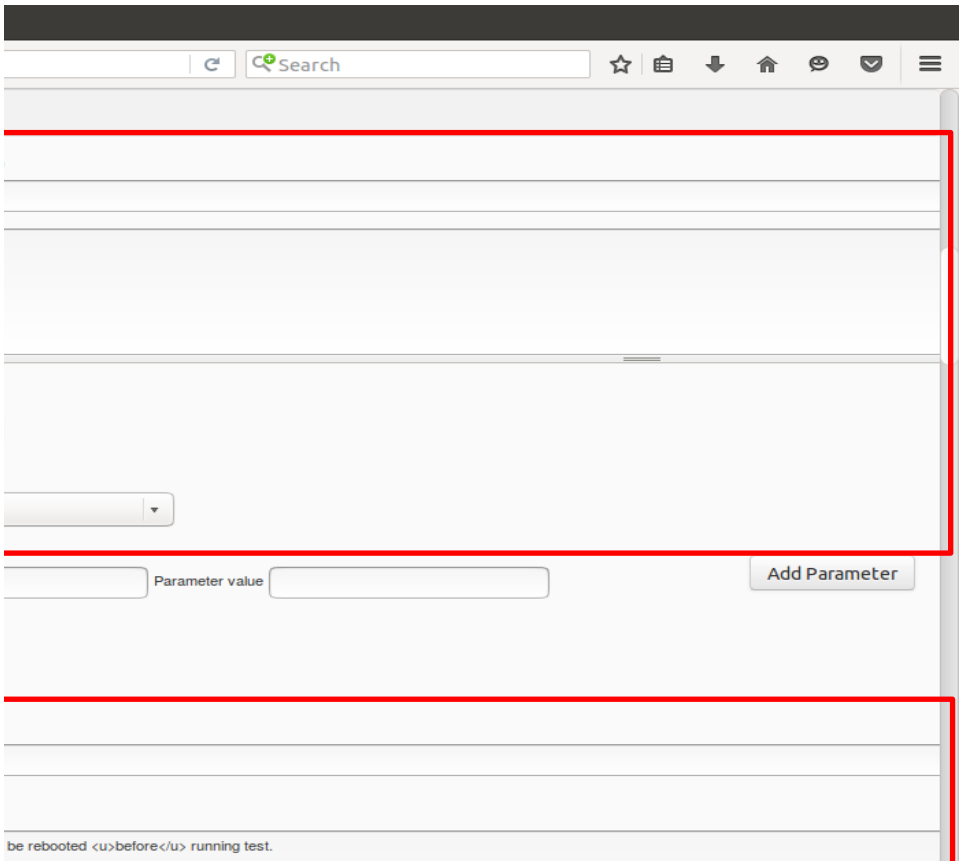


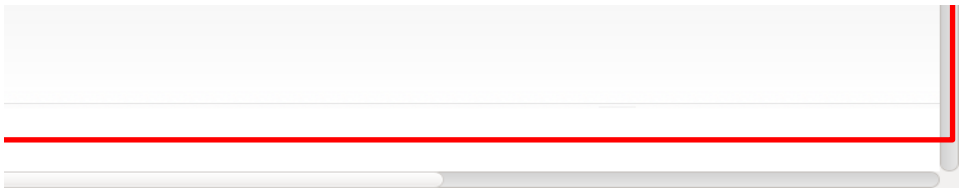


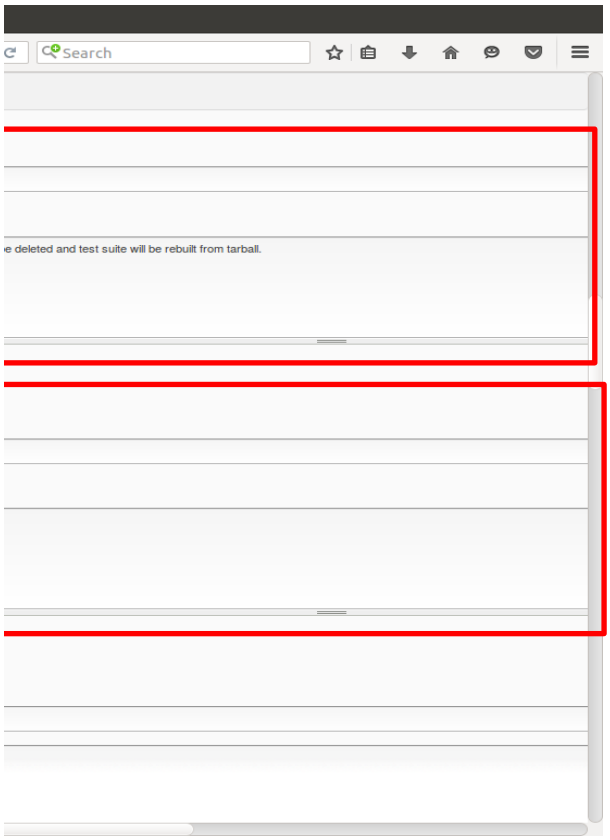


d zone in step 5)

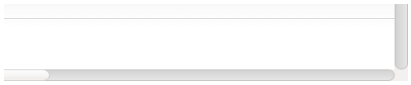
sh compression)

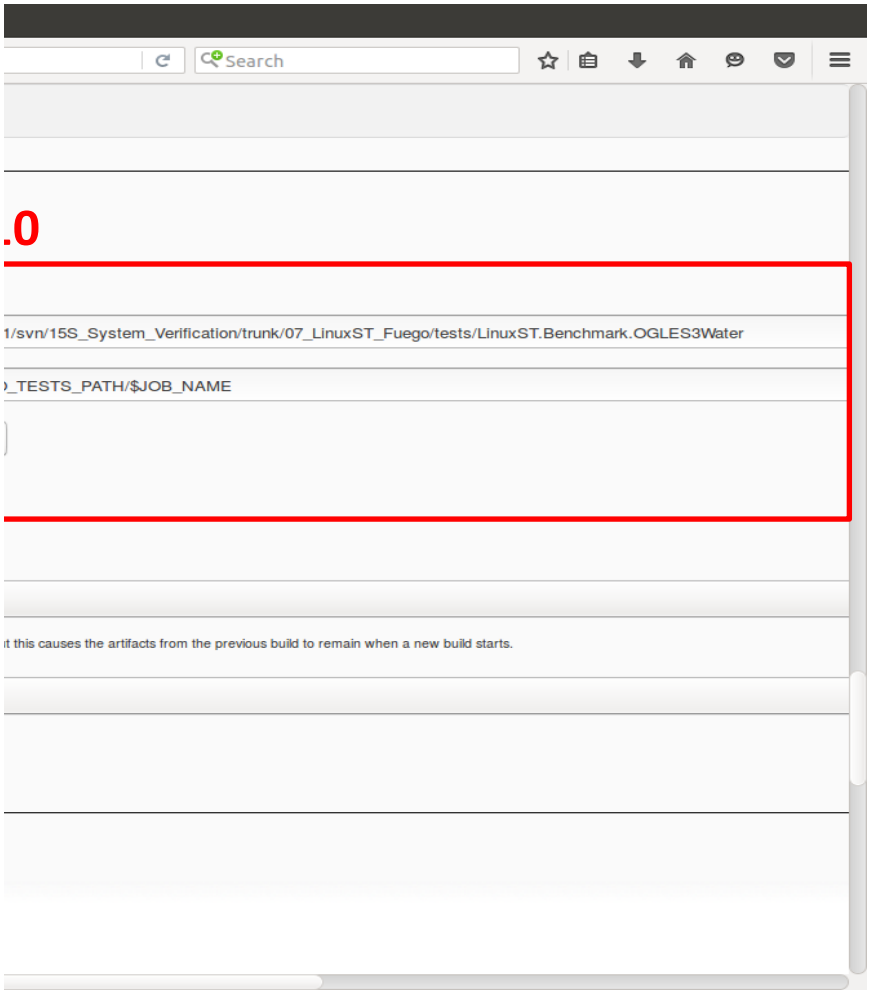




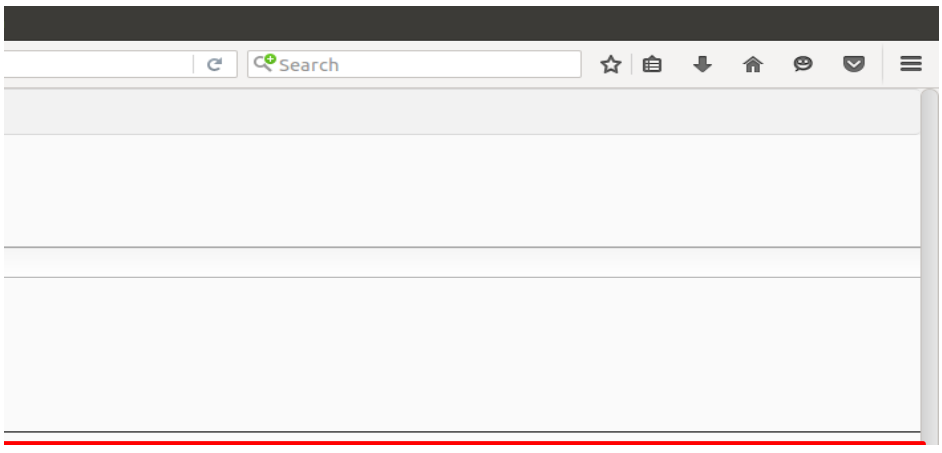

Search
e deleted and test suite will be rebuilt from tarball.

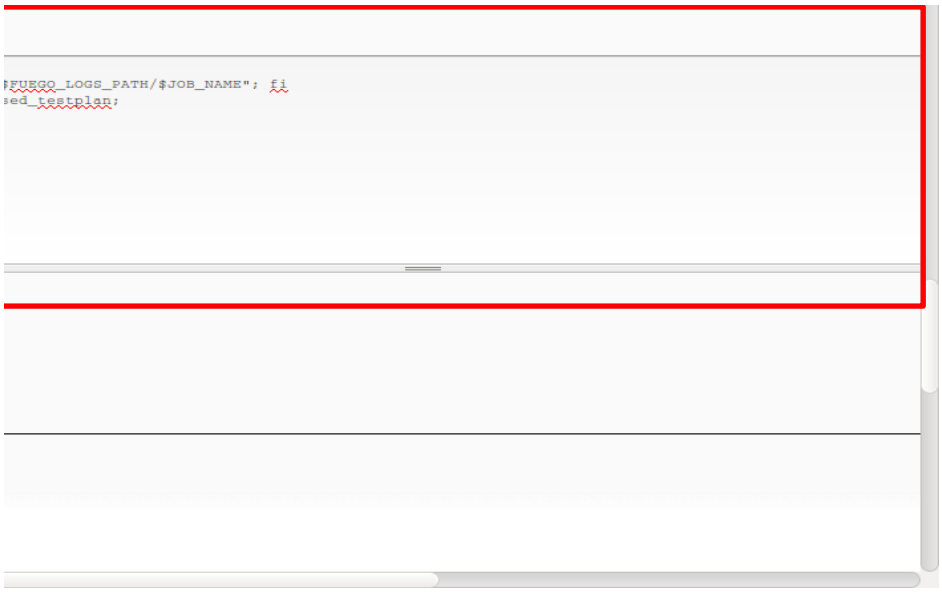

Add Parameter



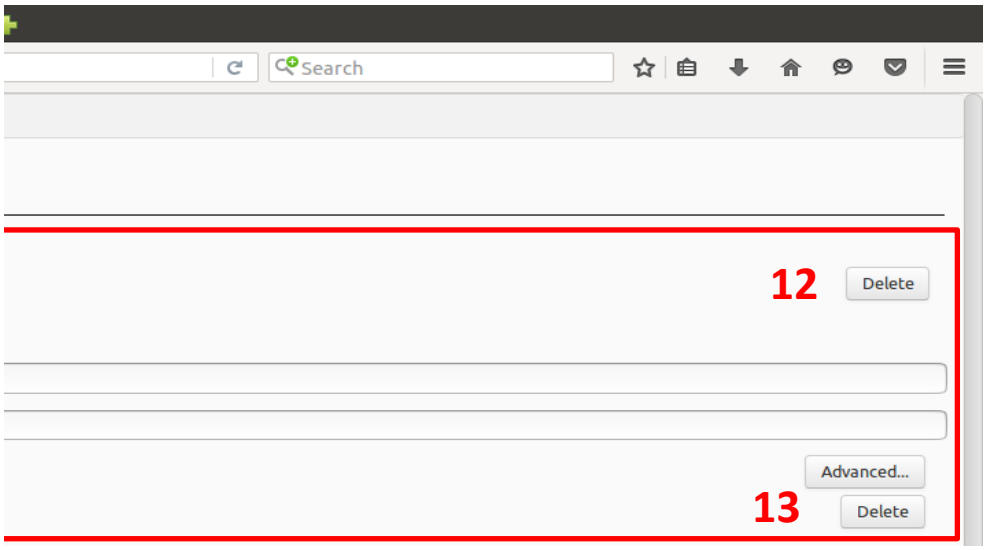


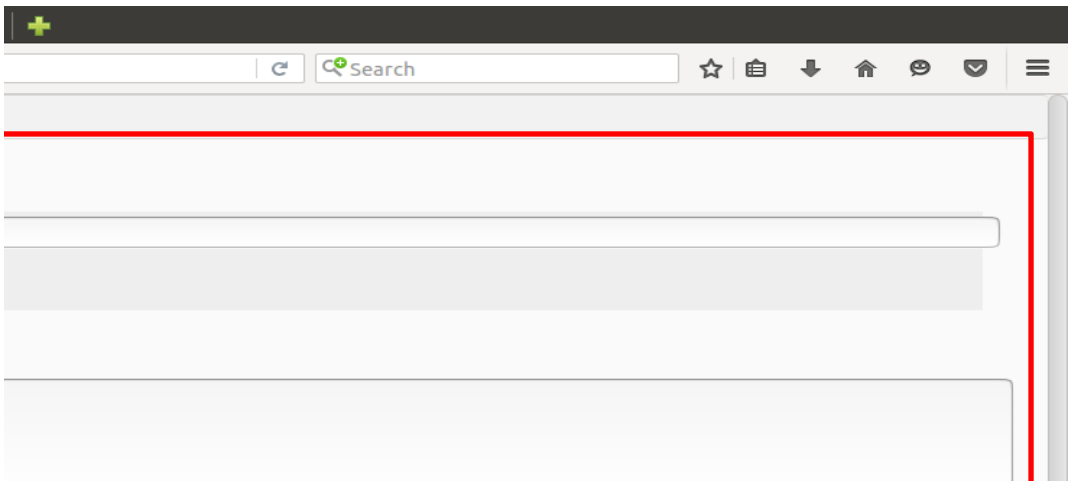
hasn't had yet. Linux FUEGO TCs is based on bash shell, so this line of script is compulsory

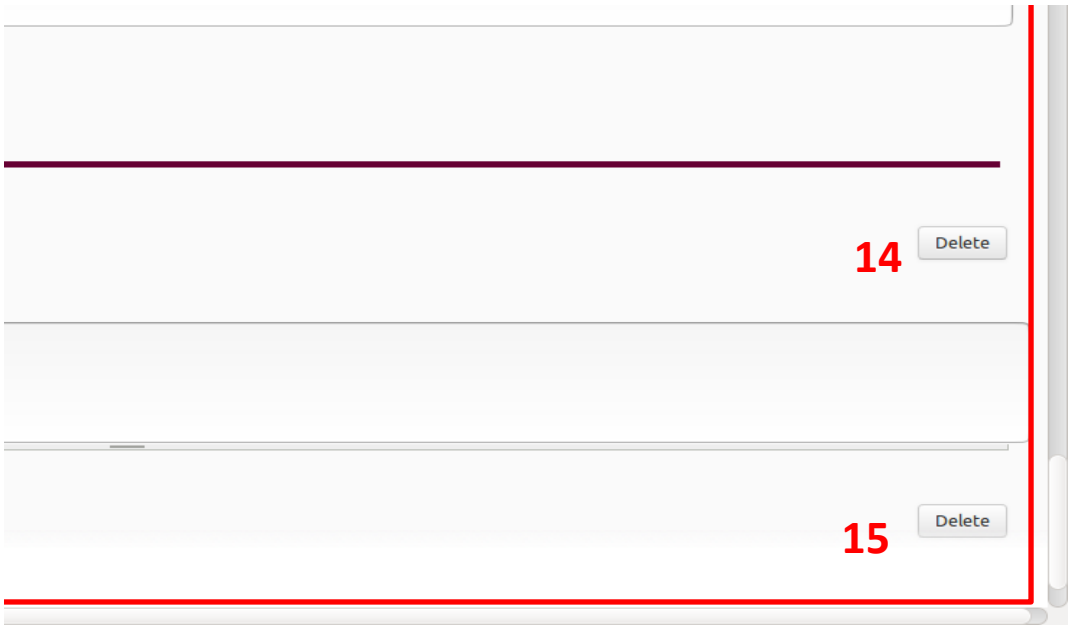



FUEGO_LOGS_PATH/$JOB_NAME"; fi
sed testplan;

supported)


Search
12
Delete
Advanced...
13
Delete

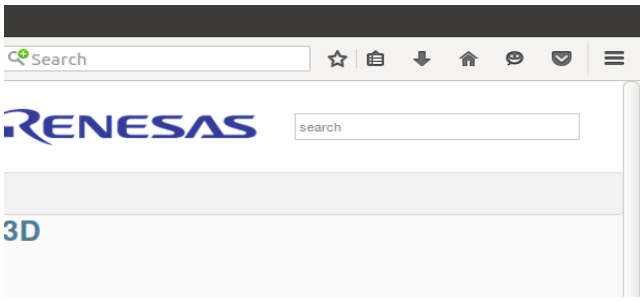

Search

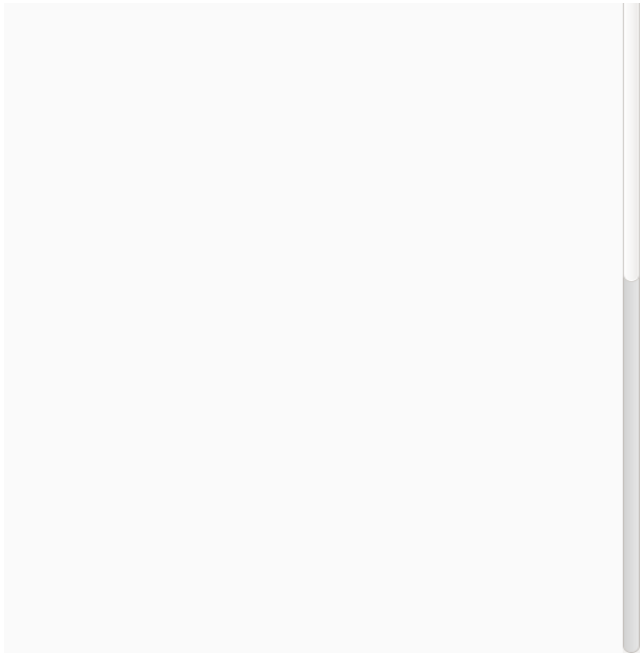


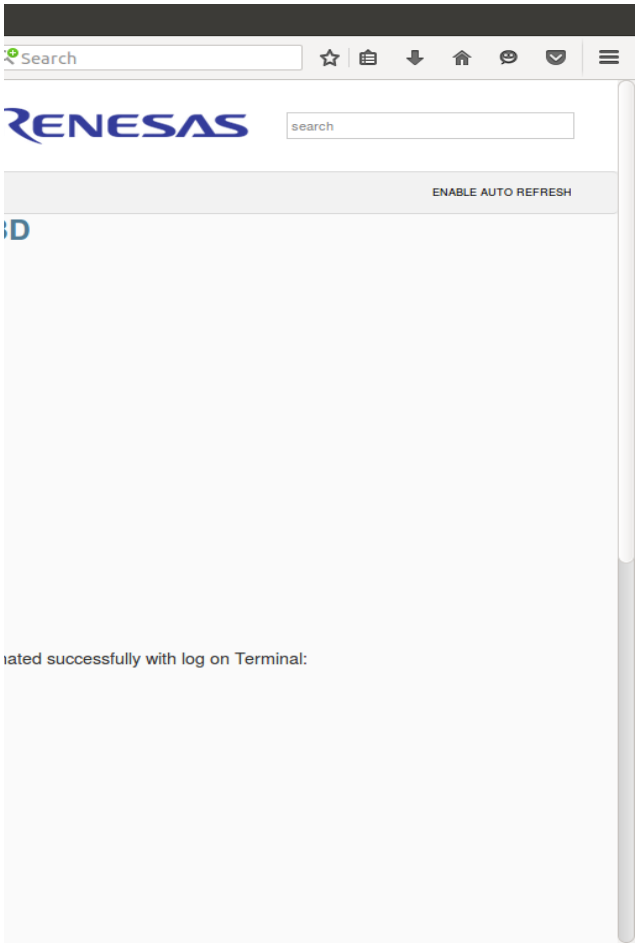
ecessary applications/libraries onto Target board automatically before executing other TCs

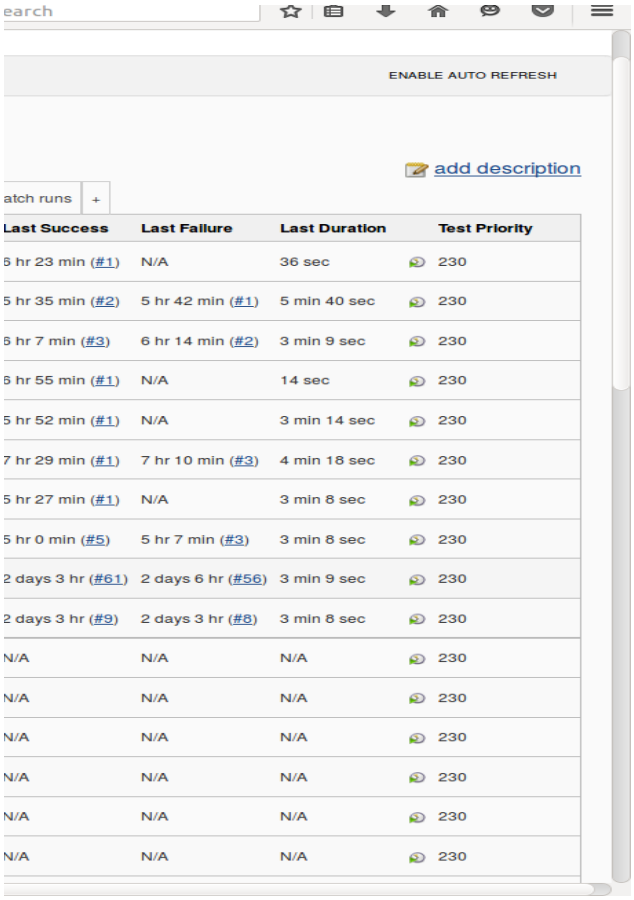
ırd

was modified, please check this option to apply the modifies





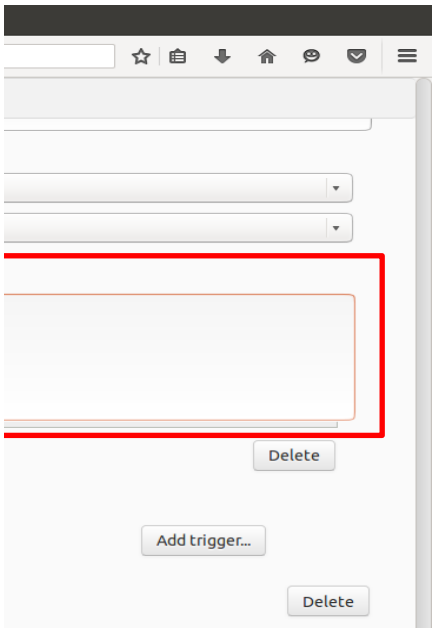

Search
RENESAS
search
ENABLE AUTO REFRESH
ated successfully with log on Terminal:

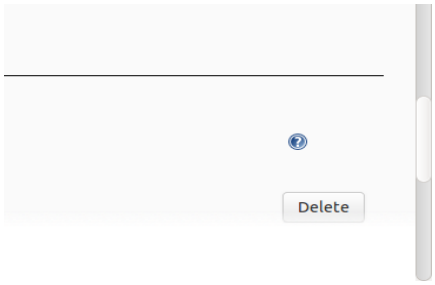


Last Success	Last Failure	Last Duration	Test Priority
5 hr 23 min (#1)	N/A	36 sec	230
5 hr 35 min (#2)	5 hr 42 min (#1)	5 min 40 sec	230
5 hr 7 min (#3)	6 hr 14 min (#2)	3 min 9 sec	230
5 hr 55 min (#1)	N/A	14 sec	230
5 hr 52 min (#1)	N/A	3 min 14 sec	230
7 hr 29 min (#1)	7 hr 10 min (#3)	4 min 18 sec	230
5 hr 27 min (#1)	N/A	3 min 8 sec	230
5 hr 0 min (#5)	5 hr 7 min (#3)	3 min 8 sec	230
2 days 3 hr (#61)	2 days 6 hr (#56)	3 min 9 sec	230
2 days 3 hr (#9)	2 days 3 hr (#8)	3 min 8 sec	230
N/A	N/A	N/A	230
N/A	N/A	N/A	230
N/A	N/A	N/A	230
N/A	N/A	N/A	230
N/A	N/A	N/A	230
N/A	N/A	N/A	230

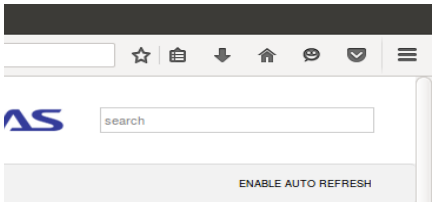
te many TCs
they want to execute all time

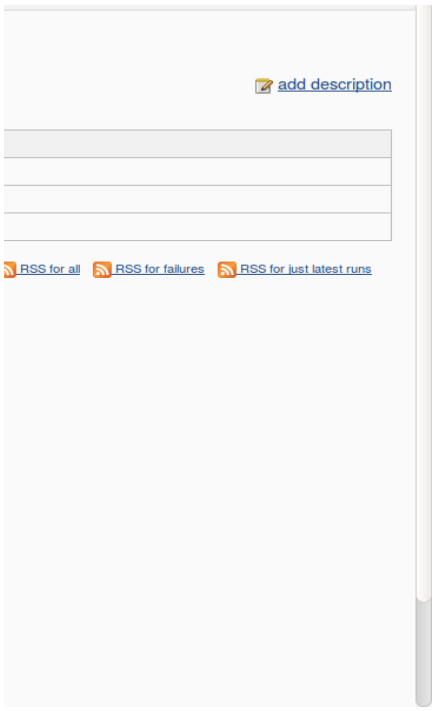
arting new TC

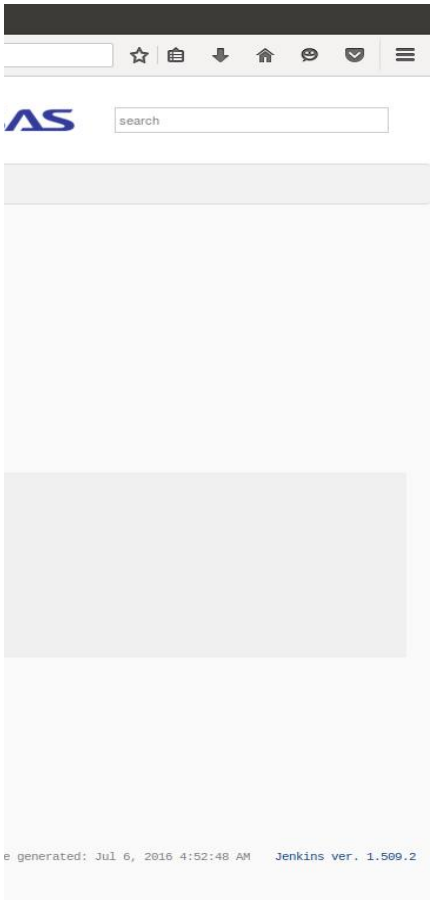




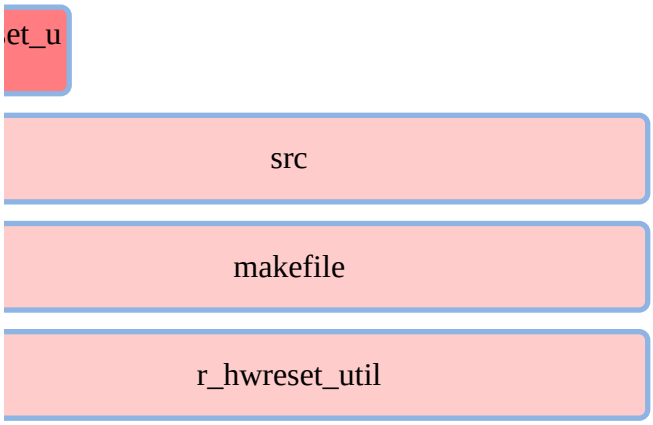
ɔt, clean-up) and Run Test




add description
RSS for all
RSS for failures
RSS for just latest runs


search
e generated: Jul 6, 2016 4:52:48 AM
Jenkins ver. 1.509.2

script to prepare data in device later


et_u
src
makefile
r_hwreset_util

ST_function.sh)

a3_ver3.x.xls

m