

# Changing Test Parameters with Gerrit Commit Messages

- [Introduction](#)
- [General Parameters](#)
- [Node Parameters](#)
- [Versions](#)
- [Test Groups](#)

## Introduction

There may be times that you wish to influence the building and testing carried out on your change. You might for example be fixing an issue that affects a particular distribution or combination of distributions, for this reason you can change the tests carried out by indicating in the commit message the changes you require. Test Parameter sessions are *in addition* to the normal tests that would be run against a patch. This allows a patch with specific or unusual testing requirements to ensure that sufficient additional testing is run to gain confidence in the change being made. For patches that are of an experimental nature (i.e. developer is not sure of functionality, or only wants a limited set of tests to be run just to try something), it is also possible to submit a patch with the [fortestonly](#) parameter.

To run additional sessions for your patch, add 'Test-Parameters: ' with space-separated name value pairs to your commit message.

For example:

```
Test-Parameters: env=ONLY=32 testlist=conf-sanity
```

This will cause Autotest to run the normal test sessions plus 1 additional session where only conf-sanity test #32 will run.

Multiple 'Test-Parameters:' lines can be defined:

```
Test-Parameters: testgroup=review-ldiskfs clientdistro=sles12sp3 serverdistro=sles12sp3
Test-Parameters: testgroup=review-ldiskfs clientdistro=ubuntu1804 serverdistro=el7
```

Long lists can be catered for by escaping the carriage return:

```
Test-Parameters: ostcount=2 clients=1 ostsizegb=2 mdssizegb=2 env=SLOW=yes \
                testlist=sanity,liblustre
```

Quotations can be used when spaces are necessary in a value:

```
Test-Parameters: testlist=sanity env=SANITY_EXCEPT="101g 102i"
```

The test parameter sessions can be influenced in many ways, see the [General Parameters](#) and [Node Parameters](#) sections below for all of the options.

## General Parameters

Below is the list of general test parameters that can be used to run custom test sessions. These parameters differ from the [node parameters](#) in that these do not need to be specified with a node type prefix.

For all of the examples below the 'Test-Parameters:' marker has been omitted to simplify the examples.

### austeroptions | String

Pass through options for auster.

Valid values: See "Auster usage help" section on the [Setting up a Lustre Test Environment](#) wiki.

```
austeroptions=-R
```

### combinedmdsmgs | Boolean

When true, the MGT will share a partition with an MDT. When false, AT will create an additional partition to be used by just the MGT.

NOTE: mdfilesystemtype will override mgfilesystemtype when combinedmdsmgs is true. If standalonemgs is set to true, this option will be ignored.

Valid values: true, false (no value is the same as true)

```
# creates a shared partition to be used by the MGT and a MDT - typically not necessary since this is the
default
combinedmdsmgs
# or
combinedmdsmgs=true

# creates an additional partition to be used by the MGT
combinedmdsmgs=false
```

### env, envdefinitions | String

Comma separated environment definitions passed to the test environment. For definitions requiring spaces, enclose them in quotations.

NOTE: Be very careful setting environment variables directly (for example OSTFSTYPE=zfs) because Autotest creates a config file based on the environment it builds. If you ask for something at odds with Autotest's expectations you will see failure instead of success. In this case, for example, you should use the filesystemtype keyword described on this page. Autotest will then create the appropriate environment variables, the same is true for other things like ostcount instead of OSTCOUNT, ostsizegb instead of OSTSIZE, etc. Only use the env parameter when a direct variable is not listed here.

```
env=SLOW=yes
envdefinitions=SLOW=yes

# multiple definitions, one with spaces that must be enclosed in quotations
env=SLOW=yes,SANITY_EXCEPT="101g 102i"
```

### failover | Boolean

Setup cluster in failover configuration

Valid values: true, false (no value is the same as true)

```
# enable failover
failover
# or
failover=true

# disable failover setup
failover=false
```

### forbuildonly | Boolean

Patch will be built by Jenkins but Autotest will not run any testing.

Valid values: true, false (no value is the same as true)

```
# mark build as for build only
forbuildonly
# or
forbuildonly=true

# disable forbuildonly - typically not necessary since the default is false
forbuildonly=false
```

### fortestonly | Boolean

The `fortestonly` parameter marks that the patch is *not intended for landing*. Any `testlist=` specified with `Test-Parameters: fortestonly` will **replace** the default tests that will be run, so it is possible to run only a subset of tests. Patches marked with `fortestonly` will **not** receive the Verified label from Maloo and **cannot be landed**. In order to land such a patch, it should be rebased and submitted **without** the `fortestonly` keyword once the patch is known to be good, or the commit comment in Gerrit should be edited to remove the `fortestonly` label which will rebuild and retest the patch and will preserve any reviews that the patch has received).

Valid values: true, false (no value is the same as true)

```
# mark build as for test only
fortestonly
# or
fortestonly=true

# mark patch as for test only and run a custom list of test
fortestonly testlist=conf-sanity

# disable fortestonly - typically not necessary since the default is false
fortestonly=false
```

### fstype | String

Sets the file system type for all server nodes (MDS, MGS and OSS).

NOTE: `fstype` should not be used in combination with `<mdt/mgt/ost>filesystemtype`.

Valid values: ldiskfs, zfs

```
# set all server nodes to use zfs
fstype=zfs

# set all server nodes to use ldiskfs
fstype=ldiskfs

# this may result in an unexpected configuration
fstype=zfs mdtfilesystemtype=ldiskfs
```

### iscsi | boolean

Use iSCSI for failover testing

Valid values: 0 (no iSCSI), 1 (iSCSI)

```
# do not setup iscsi
iscsi=0

# setup iscsi
iscsi=1
```

### mdssizegb | Integer

Sets the size of the partitions on the MDS in GB.

Valid values: > 0 and must be an integer. When setting this value, take into consideration your mdcount value and that most test nodes have ~90GB of disk space.

```
# configure 2GB partitions on the MDS
mdssizegb=2
```

### mdtcount | Integer

The total number of MDTs across all of the MDSs.

Valid values: 1 - 4 per MDS

```
# create 4 MDTs per MDS
mdscount=2 mdtcount=8

# create 1 MDT per MDS
mdscount=4 mdtcount=4
```

### mdfilesystemtype | String

Configure the file system type to use on the MDTs.

NOTE: <mdt/mgt/ost>filesystemtype should not be used in combination with fstype.

Valid values: ldiskfs, zfs

```
# configure the MDTs to use zfs
mdfilesystemtype=zfs

# configure the MDTs to use ldiskfs
mdfilesystemtype=ldiskfs

# this may result in an unexpected configuration
fstype=zfs mdfilesystemtype=ldiskfs
```

### mgssizegb | Integer

Sets the size of the partition on the MGS in GB. This value is only applicable when the session is using a stand alone MGS.

Valid values: > 0 and must be an integer. When setting this value, take into consideration that most test nodes have ~90GB of disk space.

```
# configure a 2GB partition on the MGS
mgssizegb=2
```

### mgfilesystemtype | String

Configure the file system type to use on the MGTs.

NOTE: The default value for [combinedmdsmgs](#) is true, therefore it must be set to false in order to configure a different file system type for the MGT. Also, <mdt/mgt/ost>filesystemtype should not be used in combination with [fstype](#).

Valid values: ldiskfs, zfs

```
# configure the MGTs to use zfs
mgfilesystemtype=zfs combinedmdsmgs=false

# configure the MGTs to use ldiskfs
mgfilesystemtype=ldiskfs combinedmdsmgs=false

# since combinedmdsmgs defaults to true, mgfilesystemtype will be ignored and
# the MGT will use ldiskfs
mdtfilesystemtype=ldiskfs mgfilesystemtype=zfs

# do not do this - may result in an unexpected configuration
fstype=zfs mgfilesystemtype=ldiskfs
```

### optional | Boolean

Marks the test session as optional: does not impact the verified value from Maloo and is only run if resources are immediately available.

Valid values: true, false (no value is the same as true)

```
# mark a session as optional
optional
# or
optional=true

# mark a session as required - typically not necessary since false is the default
optional=false
```

### ostcount | Integer

The total number of OSTs across all of the OSSs.

Valid values: 1 - 4 per OSS

```
# create 4 OSTs per OSS
osscount=2 ostcount=8

# create 1 OST per OSS
osscount=4 ostcount=4
```

### ostfilesystemtype | String

Configure the file system type to use on the OSTs.

NOTE: <mdt/mgt/ost>filesystemtype should not be used in combination with [fstype](#).

Valid values: ldiskfs, zfs

```
# configure the OSTs to use zfs
ostfilesystemtype=zfs

# configure the OSTs to use ldiskfs
ostfilesystemtype=ldiskfs

# do not do this - may result in an unexpected configuration
fstype=zfs ostfilesystemtype=ldiskfs
```

### ostsizegb | Integer

Sets the size of the partitions on the OSS in GB.

Valid values: > 0 and must be an integer. When setting this value, take into consideration your ostcount value and that most test nodes have ~90GB of disk space.

```
# configure 2GB partitions on the OSS
ostsizegb=2
```

### resumeaftercrash | Boolean

When true, Autotest will continue the session after a suite crashes. The session will continue with the suite following the one that crashed. When false, Autotest will stop the session and upload the results to Maloo immediately.

Valid values: true, false (no value is the same as true)

```
# configure the session to continue after a suite crashes - typically not necessary since the default is true
resumeaftercrash
# or
resumeaftercrash=true

# configure the session to stop after a suite crashes
resumeaftercrash=false
```

### signofftest | Boolean

Marks the session as enforcing: is used to determine the verified value from Maloo.

Valid values: true, false (no value is the same as true)

```
# mark the session as enforcing - typically not needed since the default is true
signofftest
# or
signofftest=true

# mark the session as not enforcing
signofftest=false
```

### standalonemgs | Boolean

Provisions a separate test node as the MGS. The MGS will be setup with the same parameters as the MDS unless they're overwritten.

Valid values: true, false (no value is the same as true)

```
# configure the session to run with a stand alone MGS
standalonemgs
# or
standalonemgs=true

# configure the session to run with the MGS on the MDS but separate partitions
standalonemgs=false combinedmdsmgs=false

# configure the session to run with the MGS on the MDS and have the MGT share
# a partition with a MDT
standalonemgs=false combinedmdsmgs=true

# configure the session to run with a stand alone MGS with a different file
# system type than the OSS and MDS
standalonemgs mgfilesystemtype=zfs mdfilesystemtype=ldiskfs ostfilesystemtype
```

### testgroup | String

Configure the test session to run a specific test group. Specifying a test group makes it easy to run a typical test grouping with small modifications since the session will inherit all of the values from the base test with any overrides applied (see the code block below for examples).

NOTE: testgroup can be combined with [testlist](#) to run a test group plus additional suites

Valid values: See the [Test Groups](#) section for a complete list

```
# run a typical review-ldiskfs session with a stand alone MGS
testgroup=review-ldiskfs standalonemgs

# run failover with a specific server version
testgroup=failover serverversion=2.11.50

# run review-dne-part-1 with additional suites
testgroup=review-dne-part-1 testlist=sanity-lfsck,sanity-sec
```

### testlist | String

Configure the test session to run a specific list of suites. testlist can be combined with [testgroup](#) to run a test group plus additional suites.

Valid values: sanity, sanityn, conf-sanity, mmp, replay-single, replay-dual, lnet-selftest, etc.

```
# run a custom list of suites
testlist=sanity,sanity-sec,sanity-hsm

# run review-dne-part-1 with additional suites
testlist=sanity-lfsck,sanity-sec testgroup=review-dne-part-1
```

### timeoutreprovision | Boolean

Configures the session to re-provision all of the nodes instead of simply reboot them.

Valid values: true, false (no value is the same as true)

```
# re-provision nodes after a timeout
timeoutreprovision
# or
timeoutreprovision=true

# reboot the nodes after a timeout - typically not necessary since reboot is the default
timeoutreprovision=false
```

### trivial | Boolean

The `trivial` keyword can be used to reduce the testing time (both wall-clock time as well as total test system hours) for patches that do not affect code functionality, such as changes to whitespace, comments, man pages, and test scripts (in conjunction with the `testlist` keyword to ensure the modified test script is run if it isn't already). Patches marked with `trivial` in the `Test-Parameters: list` will run the `review-ldiskfs` test session **instead** of the regular tests, so it currently runs only `sanity` and `lnet-selftest` to ensure basic functionality. If changes are only being made to a test script outside of those already run by default, an additional `testlist=<test-script(s)>keyword` should be added with a comma-separated list of modified tests to ensure they are run to validate the changes. The `trivial` keyword will reduce the test completion time from approximately 10h elapsed and 30h of total test system time to approximately 3h elapsed/system time. This saves test system resources that may be better spent on other patches.

Valid values: true, false (no value is the same as true)

```
# mark the patch as trivial
trivial
# or
trivial=true

# mark the patch as non-trivial - typically not necessary since this is the default
trivial=false
```

## Node Parameters

Node parameters are used to change how specific node types are configured. They must be prefixed with the node type being changed. Valid node types are **client**, **mds**, **mgs**, **oss** and **server**. Server is an alias that allows users to modify a value for all server node types (mds, mgs and oss). For example instead of writing

```
mdsdistro=e17 mgsdistro=e17 ossdistro=e17
```

Users can simply write

```
serverdistro=e17
```

### arch | String or Array

Sets the architecture for the specified node type

Valid values: x86\_64, ppc64, aarch64 (architecture must have been built for patch)

```
# clients will use x86_64
clientarch=x86_64
```

### buildno | Integer

Used in conjunction with [job](#) to install a specific build on the specified node. [job](#) must be specified with buildno and [version](#) cannot be specified with buildno.

Valid values: Any valid Jenkins build number for the specified job

```
# install the server with a specific job/build combination
serverbuildno=343 serverjob=lustre-master

# this would fail since serverjob is not specified
serverbuildno=343
```

### count | Integer

The number of nodes to use for the specified node type.

NOTE: For MDS and OSS nodes, it's best to also set [mdscount](#) / [ostcount](#) to ensure you have the expected number of targets.

Valid values:

clientcount	2 - 4
mdscount	1 - 6
osscount	1 - 4

```
# configure session to run with 4 clients, 2 MDSs and 1 OSS
clientcount=4 mdscount=2 osscount=1
```

### distro | String

Distribution to use for the specified node type.

Valid values: el7, el7.5, el7.6, ubuntu1604, ubuntu1804, sles12sp3, etc. (The distribution specified must have been built for the patch being tested. The distros built can be seen on the patch's build page in [Jenkins](#).)

```
# specify the client to use el7 and all servers sles12sp3
clientdistro=el7 serverdistro=sles12sp3

# specify the client to use ubuntu1804, MDS to use el7 and OSS to use sles12sp3
clientdistro=ubuntu1804 mdsdistro=el7 ossdistro=sles12sp3
```

### ibstack | String

The IB stack to use for the specified node type.

Valid values: inkernel, ofa (The IB stack type must have been built for the build being tested. The stack types can be seen on the patch's build page in [Jenkins](#).)

```
clientibstack=inkernel
```

### job | String

Used with buildno to install a specific build on the specified node. [buildno](#) must be specified with job. version cannot be specified with job.

Valid values: Any valid Jenkins job, such as lustre-reviews, lustre-master, etc.

```
# install the server with a specific job/build combination
serverjob=lustre-b2_10 serverbuildno=123

# this would fail since buildno is not specified for client
serverjob=lustre-b2_10 serverbuildno=123 clientjob=lustre-master
```

### selinux | Boolean

Enables selinux on the specified node type.

Valid values: true, false (no value is the same as true)

```
# enables selinux on clients
clientselinux
# or
clientselinux=true

# disables selinux on clients - not typically necessary since false is the default
clientselinux=false
```

### version | String

The version of Lustre to use for the specified node type.

NOTE: version cannot be specified with job and build

Valid values: For a list of valid versions see the [Versions](#) section.

```
# specify patch level versions
clientversion=2.10.3 serverversion=2.11.0

# specify minor versions which uses the highest patch level version
clientversion=2.10 serverversion=2.11
```

## Versions

Versions are pointers to job/build combinations and simplify using a specific Lustre version on a test node. Versions can be specified in the test parameters using the [version](#) node parameter. If there is a version missing send an email to [charlie@whamcloud.com](mailto:charlie@whamcloud.com)

Version	Build
2.7.0 (2.7)	<a href="https://build.whamcloud.com/job/lustre-b2_7/29">https://build.whamcloud.com/job/lustre-b2_7/29</a>
2.8.0 (2.8)	<a href="https://build.whamcloud.com/job/lustre-b2_8/12">https://build.whamcloud.com/job/lustre-b2_8/12</a>
2.9.0 (2.9)	<a href="https://build.whamcloud.com/job/lustre-b2_9/22">https://build.whamcloud.com/job/lustre-b2_9/22</a>
2.10.0	<a href="https://build.whamcloud.com/job/lustre-b2_10/5">https://build.whamcloud.com/job/lustre-b2_10/5</a>
2.10.1	<a href="https://build.whamcloud.com/job/lustre-b2_10/30">https://build.whamcloud.com/job/lustre-b2_10/30</a>
2.10.2	<a href="https://build.whamcloud.com/job/lustre-b2_10/52">https://build.whamcloud.com/job/lustre-b2_10/52</a>
2.10.3	<a href="https://build.whamcloud.com/job/lustre-b2_10/69">https://build.whamcloud.com/job/lustre-b2_10/69</a>
2.10.4	<a href="https://build.whamcloud.com/job/lustre-b2_10/116">https://build.whamcloud.com/job/lustre-b2_10/116</a>
2.10.5	<a href="https://build.whamcloud.com/job/lustre-b2_10/136">https://build.whamcloud.com/job/lustre-b2_10/136</a>
2.10.6	<a href="https://build.whamcloud.com/job/lustre-b2_10/152">https://build.whamcloud.com/job/lustre-b2_10/152</a>
2.10.7	<a href="https://build.whamcloud.com/job/lustre-b2_10/168/">https://build.whamcloud.com/job/lustre-b2_10/168/</a>
2.10.8 (2.10)	<a href="https://build.whamcloud.com/job/lustre-b2_10/170/">https://build.whamcloud.com/job/lustre-b2_10/170/</a>
2.11.0 (2.11)	<a href="https://build.whamcloud.com/job/lustre-b2_11/2">https://build.whamcloud.com/job/lustre-b2_11/2</a>
2.12.0	<a href="https://build.whamcloud.com/job/lustre-b2_12/2">https://build.whamcloud.com/job/lustre-b2_12/2</a>
2.12.1	<a href="https://build.whamcloud.com/job/lustre-b2_12/13">https://build.whamcloud.com/job/lustre-b2_12/13</a>
2.12.2	<a href="https://build.whamcloud.com/job/lustre-b2_12/18/">https://build.whamcloud.com/job/lustre-b2_12/18/</a>
2.12.3	<a href="https://build.whamcloud.com/job/lustre-b2_12/53/">https://build.whamcloud.com/job/lustre-b2_12/53/</a>
2.12.4	<a href="https://build.whamcloud.com/job/lustre-b2_12/66/">https://build.whamcloud.com/job/lustre-b2_12/66/</a>
2.12.5 (2.12)	<a href="https://build.whamcloud.com/job/lustre-b2_12/87">https://build.whamcloud.com/job/lustre-b2_12/87</a>
2.13.0 (2.13)	<a href="https://build.whamcloud.com/job/lustre-b2_13/6/">https://build.whamcloud.com/job/lustre-b2_13/6/</a>

## Test Groups

Test groups are set lists of Lustre test suites managed by Autotest.

Name	Suites
<b>basic</b>	conf-sanity
<b>development</b>	metadata-updates
<b>failover</b>	recovery-mds-scale, recovery-random-scale, recovery-double-scale, recovery-small, replay-ost-single, replay-dual, replay-vbr, mmp, replay-single
<b>full</b>	Inet-selftest, runtests, sanity, sanity-scrub, sanity-benchmark, sanity-lfsck, sanityn, sanity-hsm, sanity-flr, sanity-dom, sanity-ls-snapshot, conf-sanity, racer, replay-single, recovery-small, replay-ost-single, replay-dual, replay-vbr, insanity, sanity-quota, sanity-sec, sanity-pfl, lustre-rsync-test, metadata-updates, ost-pools, mds-survey, mmp, performance-sanity, parallel-scale, large-scale, obdfilter-survey, parallel-scale-nfsv3, parallel-scale-nfsv4, posix
<b>full-patchless</b>	Inet-selftest, runtests, sanity, sanity-scrub, sanity-benchmark, sanity-lfsck, sanityn, sanity-hsm, sanity-flr, sanity-dom, sanity-ls-snapshot, racer, insanity, sanity-quota, sanity-sec, sanity-pfl, lustre-rsync-test, metadata-updates, ost-pools, mds-survey, mmp, performance-sanity, parallel-scale, large-scale, obdfilter-survey, parallel-scale-nfsv3, parallel-scale-nfsv4, posix

<b>regression-mpi</b>	performance-sanity, large-scale, parallel-scale, sanity-lnet
<b>review</b>	lnet-selftest, runtests, sanity, sanityn, replay-single, conf-sanity, recovery-small, replay-ost-single, insanity, sanity-quota, lustre-rsync-test, ost-pools, sanity-lfsck, sanity-hsm
<b>review-dne-part-1</b>	sanity, sanityn, recovery-small, lustre-rsync-test
<b>review-dne-part-2</b>	runtests, ost-pools, sanity-hsm, sanity-scrub, sanity-lfsck, sanity-sec, mds-survey, large-scale
<b>review-dne-part-3</b>	conf-sanity
<b>review-dne-part-4</b>	mmp, replay-single, sanity-pfl, sanity-flr, sanity-dom, sanity-quota, replay-ost-single, insanity
<b>review-dne-selinux</b>	sanity, recovery-small, sanity-selinux
<b>review-dne-selinux-ssk</b>	sanity, recovery-small, sanity-selinux, sanity-sec
<b>review-dne-ssk</b>	sanity, recovery-small, sanity-sec
<b>review-dne-zfs</b>	runtests, sanity, sanityn, replay-single, conf-sanity, recovery-small, replay-ost-single, insanity, sanity-quota, lustre-rsync-test, ost-pools, sanity-lfsck, sanity-hsm
<b>review-dne-zfs-part-1</b>	sanity, sanityn, recovery-small, lustre-rsync-test
<b>review-dne-zfs-part-2</b>	runtests, ost-pools, sanity-hsm, sanity-scrub, sanity-lfsck, sanity-sec, mds-survey, large-scale
<b>review-dne-zfs-part-3</b>	conf-sanity
<b>review-dne-zfs-part-4</b>	mmp, replay-single, sanity-pfl, sanity-flr, sanity-dom, sanity-quota, replay-ost-single, insanity
<b>review-ldiskfs</b>	lnet-selftest, sanity
<b>review-ldiskfs-arm</b>	lnet-selftest, sanity
<b>review-ldiskfs-dne</b>	lnet-selftest, sanity
<b>review-ldiskfs-ubuntu</b>	lnet-selftest, sanity
<b>review-zfs</b>	sanity-quota, sanity-flr, sanity-lsnapshot, replay-single, replay-ost-single, insanity

<b>review-zfs-part-1</b>	runtests, sanity, sanityn, sanity-quota, ost-pools, sanity-lfsck, sanity-hsm, sanity-flr
<b>review-zfs-part-2</b>	replay-single, conf-sanity, recovery-small, replay-ost-single, insanity, lustre-rsync-test, large-scale, mds-survey
<b>tiny</b>	sanity, mmp