

Code Submission Process - RDK Central Gerrit

- [Create a JIRA ticket](#)
- [Clone the Repository](#)
- [Work on the change, commit to local clone](#)
- [Push the new changes for Gerrit for review](#)
- [Review notifications and addition of new reviewers](#)
- [Scan and build on code submission](#)
- [Code review and scoring process](#)
- [Submit code change](#)
- [Abandon change](#)
- [Submitted, Merge Pending](#)
- [Change needs to be reworked](#)
- [Gerrit merge failure as a result of a conflict](#)
- [RDK Components - Product Branch](#)

In order to contribute code, first-time users are requested to agree to the license at <https://wiki.rdkcentral.com/signup.action>.

RDK components are hosted at code.rdkcentral.com. You can submit your code changes for review via that site using the workflow outlined below.

Create a JIRA ticket

- Refer to [JIRA Guidelines](#) for creating a JIRA before pushing your code changes in code.rdkcentral.com

Clone the Repository

Clone the component repository from the Gerrit server <https://code.rdkcentral.com/r/> into a local workspace

Clone with commit-msg hook (to add Change-ID footer to commit messages)

```
git clone https://code.rdkcentral.com/r/<component-name> <component-name> -b <branch-name>
cd <component-name>
gitdir=$(git rev-parse --git-dir); curl -o ${gitdir}/hooks/commit-msg https://code.rdkcentral.com/r/tools/hooks/commit-msg ; chmod +x ${gitdir}/hooks/commit-msg
```

[Click here](#) to find the details about <component-name> & <branch-name> for code submission .

Note: The commit-msg hook is installed in the local Git repository and is a prerequisite for Gerrit to accept commits. The inclusion of the unique Change-ID in the commit message allows Gerrit to automatically associate a new version of a change back to its original review.

Note: You may need to configure your Git identity on the cloned repository. The email address that your local Git uses should match the email address listed in Gerrit.



- Cloning the code before login once to code.rdkcentral.com, user would get the Authentication error, even though the account is in good standing and has all the required access.
- Please login to code.rdkcentral.com before attempting to clone.

Example commands to run are as follows:

```
$ git config user.name "John Doe"
$ git config user.email "john.doe@example.org"
```

Work on the change, commit to local clone

Each commit constitutes a change in Gerrit and must be approved separately. It is recommended to squash several commits into one that represents a change to the project.

If necessary, it is possible to squash a series of commits into a single commit before publishing them, using interactive rebase:

```
$ git rebase --interactive
```

It is important to preserve the *Change-Id* line when editing and there should only be one "pick" entry at the end of this process. The end result is to submit one change to Gerrit.

Push the new changes for Gerrit for review

Commits will be BLOCKED if the format of the commit message does not comply with the standard. You will see a warning as to why the commit was blocked.

Mandatory Information in Commit Message

1. Associated JIRA ticket (Following the [Guideline](#) to create a JIRA)
2. Reason for change information
3. Test procedure by which change can be verified
4. Possible risks of failure

```
$ git commit --amend
```

<Add your commit message in below format>

```
<JIRA TICKET #1>, <JIRA TICKET #2>, <JIRA TICKET #n> : <one line summary of change>
<empty line>
Reason for change: <explanation of change>
Test Procedure: < test procedure>
Risks: <side effects and other considerations> [Note: state None if there are no other considerations]
<empty line>
Signed-off-by: Your Name <your_name@email.com>
```

Submit your code changes for review

```
$ git push origin HEAD:refs/for/<branch>
```

When interfacing with Gerrit you push to a virtual branch `/refs/for/<branch>`, representing "code review before submission to branch". Gerrit will subsequently assign a unique URL for the change, to facilitate access and review via the web UI.

Notes:

- *HEAD* is a Git symbolic reference to the most recent commit on the current branch. When you change branches, *HEAD* is updated to refer to the new branch's latest commit.
- The *refspec* in the git push operation takes the form ***source:destination*** (***source*** is the local ref being pushed, ***destination*** is the remote ref being updated).

Review notifications and addition of new reviewers

Component owners/reviewers/approvers, defined as specific groups in Gerrit, will be added to the review by default. You may request additional feedback by specifically adding reviewers via the Gerrit web GUI.

Scan and build on code submission

BlackDuck, copyright scanning and build jobs will be triggered automatically from CMF Jenkins. The output of these jobs is integrated into the Gerrit voting process via custom labels and will reflect any 'red flag' in a file that has new code changes, whether introduced in the new change/patch-set or not. Scans will post any findings as comments in the Gerrit review. Build jobs also do that, but in addition will upload the build log to the corresponding JIRA ticket (if there is one) as an attachment.

Code review and scoring process

Reviewers can comment on and score a given change.

The default set of rules for enabling a code change for submission requires:

- a Code Review score of +2; this can only be provided by the component owner or an admin;
- +1 score on any mandatory Gerrit labels configured for the project.

The result of the scoring process and validation rules is to enable the *Submit* action on the Gerrit Web UI and subsequent merge capability to the target branch.

Label: Code Review(Highlighted in yellow color) For a change to be mergeable, the latest patch set must have a '+2' value approval in this category or label, and no '-2 Do not submit'. Thus -2 on any patch set can block a submit, while +2 on the latest patch set enables it for merging.

Labels: Blackduck/Copyright/Component-Build (Highlighted in yellow color) For a change to be mergeable, the change must have a '+1' score on these labels, and no '-1 Fails'. Thus, '-1 Fails' can block a submit, while '+1' enables a submit.

Change 24009 - Merged

RDK-20123, DELIA-32631 : btmgr crashes on using btrMgrTest to pair a device

Reason for change: If a Pair-Unpair is performed within the 3-min flushOut window of Disc devices, then BTR stack will be keeping a stale entry of the device which was Paired-Unpaired. This results in crash if we further use it to pair by btrMgrTest App. Test Procedure: btrMgrTest - Start Disc - Pair_a_Dev - Unpair_Dev, after unpairing try to pair the same dev - check for crash. Risks: Low

Signed-off-by: Prasanth Rayan <prashanth_rayan@comcast.com> (cherry picked from commit 92803c80dadd1ecaf6c1f6d730e9f3862398bfee) (cherry picked from commit 0950962a3d620e7ec5bda3d188323a94b652ba9b)

Change-Id: Ib9697069ba880de6282a2185c2086332b4419949

Author: Prasanth Rayan <prashanth_rayan@comcast.com> Feb 18, 2019 5:51 PM
Committer: rdkcmf-scm-jenkins2 <rdkcmf-scm-jenkins2@code.rdkcentral.local> Apr 5, 2019 1:57 AM
Commit: c39e4fba88ce21786b8e25e92bb3578e740c1b0 (browse)
Parent(s): d3a56878192bec3f9539fa6bf9ba647bdf4ea369 (browse)
Change-Id: Ib9697069ba880de6282a2185c2086332b4419949

Owner: Stephen Barrett
Assignee: Jenkins
Reviewers: Jenkins, Stephen Barrett
Project: rdk/components/generic/bluetooth
Branch: rdk-next
Topic: cmf-update-RDK-20123-stable2-20190404
Updated: 3 days ago

Cherry Pick Revert Follow-Up Ignore

Mark Reviewed

Related Changes (2) Cherry-Picks (1)

DELIA-33022 : Fix TileID Retrieval w/wo Audio play

RDK-20123, DELIA-32631 : btmgr crashes on using btrMgr...

Blackduck +1 protex_scan
CMF-Approval +1 Stephen Barrett
Code-Review +2 Stephen Barrett
Component-Build +1 Jenkins
Copyright +1 protex_scan
Review-State +11 Jenkins
Verified +10 Stephen Barrett

Files

Open All Diff against Base

File Path	Comments	Size
Commit Message	101	
src/btrCore.c	+68, -33	



Review input is generally referred to as labelling with a positive/negative score.

Submit code change

Only authorized users, i.e. component owners, component approvers or admins, can submit the change allowing Gerrit to merge it to the target branch as soon as possible. A change can be submitted, having satisfied the approval conditions described earlier, by clicking the 'Submit Patch Set n' button within the Gerrit UI. When a change has been Submitted, it is automatically merged to the target branch by Gerrit.

Abandon change

Depending on the review outcome, it might be decided to abandon the change. The component owner or an authorised user may abandon the change by clicking the "Abandon Change" button. The abandoned changes are not removed from the Gerrit database and can be restored at a later stage.

Submitted, Merge Pending

If a change depends on another change that is still in review, it will enter this state. It will be merged automatically by Gerrit once all its dependencies are submitted and merged.

Change needs to be reworked

If you need to rework a change, you need to push another commit with the same *Change-ID* as the original in its commit message. This is the mechanism Gerrit uses to associate or link the two items. The `--amend` option to the Git commit command prevents a new *Change-ID* being generated by the *commit-msg* hook.

The basic steps are outlined below.

First, fetch the change. If you still have the checkout that was used to push the original change, you can skip this step.

```
$ git fetch https://user@code.rdkcentral.com/r/component1 refs/changes/02/2/1 && git checkout FETCH_HEAD
```

where the numbering scheme for fetching the changes is as follows:

refs/changes/<last two digits of change number> <change number> <patch set number>



Gerrit will specify this fetch URL via the web UI on the 'Download' link on the review page for the change in question, you just paste it into the command line.

The screenshot shows the Gerrit web interface for a change review. The top navigation bar includes links for 'All', 'My', 'Projects', 'People', 'Documentation', 'Changes', 'Draft Comments', 'Edits', 'Watched Changes', 'Starred Changes', and 'Groups'. The main content area displays 'Change 24009 - Merged' with a description of the change, a list of reviewers (Steph, Jenkins, protex), and a list of files (src/btrCore.c) with their respective commit messages and sizes. The bottom section shows the 'Files' tab with a table of file paths, commit messages, and sizes.

Next, make any necessary source changes, and do:

```
$ git commit --amend
$ git push origin HEAD:refs/for/<branch>
```

A new patch set is now appended to the Gerrit review item, and this will go through the same review process as before.



- The 'change number' referenced above is different to underlying Git commit ID.
- Patch-sets are numbered (starting from 1) for each review, and incremented whenever a change is amended with another Git commit.
- *FETCH_HEAD* is a Git symbolic reference and shorthand for the head of the last branch fetched and is valid only immediately after the fetch operation.

Gerrit merge failure as a result of a conflict

Essentially this means that the remote branch has evolved since this change was started and now software conflicts with changes in the remote branch. The developer must resolve the merge conflicts in their local clone and then push another patch-set.

The process is resumed at step 4, with the important distinction of committing with the `--amend` option, once the developer pulls the latest changes. **Note:** A summary of the steps involved, assuming the local branch still exists:

Rebase the local branch to the latest state of origin/<branch>;Resolve all conflicts; Commit with the `--amend` option; Push changes to Gerrit for review. After this change a new patch set is created for the change.

Note: If the local branch no longer exists, the steps are as follows :

```
$ git fetch https://user@code.rdkcentral.com/r/rdk_component_1 refs/changes/58/58/2 && git checkout FETCH_HEAD
$ git rebase origin/<branch>
[Edit the conflicting file, cleaning up the <<<<, ==== >>>> markers surrounding the conflicting lines]
$ git add <file>
$ git commit --amend
$ git push origin HEAD:refs/for/<branch>
```

RDK Components - Product Branch

Following RDK components are hosted at code.rdkcentral.com. Follow the Instructions to submit your code changes.

Example of how to use **git clone** for meta-rdk-ext component : git clone <https://code.rdkcentral.com/r/rdk/components/generic/rdk-oe/meta-rdk-ext> -b rdk-next

Component	Product Branch	License
components/opensource/rbus	rdk-next	Apache
components/opensource/rbuscore	rdk-next	Apache
components/opensource/v4l2test	master	Apache
components/opensource/wayland-egl-icegdl	master	Apache
components/opensource/waymetric	master	Apache
components/opensource/westeros	master	Apache
manifests	rdk-next	manifests
rdk/components/generic/aamp	stable2	Apache
rdk/components/generic/aampabr	stable2	Apache
rdk/components/generic/appmanager	rdk-next	Apache
rdk/components/generic/audiocapturemgr	rdk-next	Apache
rdk/components/generic/bluetooth	rdk-next	Apache
rdk/components/generic/bluetooth_mgr	rdk-next	Apache
rdk/components/generic/breakpad_wrapper	rdk-next	Apache
rdk/components/generic/cobalt-wpe	master	cmf
rdk/components/generic/cpuprocanalyzer	rdk-next	Apache
rdk/components/generic/dca	rdk-next	Apache
rdk/components/generic/dcm	rdk-next	Apache
rdk/components/generic/devicesettings	rdk-next	Apache
rdk/components/generic/diagnostics	rdk-next	Apache
rdk/components/generic/dtcp	rdk-next	Apache
rdk/components/generic/dvb/dtv-testapp	rdk-next	Apache
rdk/components/generic/dvb/siparser	rdk-next	Apache
rdk/components/generic/gst-plugins-rdk	rdk-next	LGPL
rdk/components/generic/gst-plugins-rdk-aamp	stable2	LGPL
rdk/components/generic/hdmicec	rdk-next	Apache
rdk/components/generic/hwselftest	rdk-next	Apache

rdk/components/generic/iarmbus	rdk-next	Apache
rdk/components/generic/iarmmgrs	rdk-next	Apache
rdk/components/generic/injectedbundle	rdk-next	Apache
rdk/components/generic/ledmgr	rdk-next	Apache
rdk/components/generic/libSyscallWrapper	rdk-next	Apache
rdk/components/generic/libusbctrl	rdk-next	Apache
rdk/components/generic/lxc-container-generator	master	Apache
rdk/components/generic/media_utils	rdk-next	Apache
rdk/components/generic/netmonitor	rdk-next	Apache
rdk/components/generic/netsrvmgr	rdk-next	Apache
rdk/components/generic/rdkapps	rdk-next	Apache
rdk/components/generic/rdkat	rdk-next	Apache
rdk/components/generic/rdkbrowser	rdk-next	Apache
rdk/components/generic/rdkbrowser2	rdk-next	Apache
rdk/components/generic/rdkmediaplayer	rdk-next	Apache
rdk/components/generic/rdm	rdk-next	Apache
rdk/components/generic/rfc	rdk-next	Apache
rdk/components/generic/rmf_tools/generate_si_cache	rdk-next	Apache
rdk/components/generic/rmf_tools/tenableHDCP	rdk-next	Apache
rdk/components/generic/rne	rdk-next	Apache
rdk/components/generic/sys_mon_tools/analyzers/scripts/host	rdk-next	Apache
rdk/components/generic/sys_mon_tools/analyzers/scripts/target	rdk-next	Apache
rdk/components/generic/sys_mon_tools/iarm_event_sender	rdk-next	Apache
rdk/components/generic/sys_mon_tools/iarm_query_powerstate	rdk-next	Apache
rdk/components/generic/sys_mon_tools/iarm_set_powerstate	rdk-next	Apache
rdk/components/generic/sys_mon_tools/key_simulator	rdk-next	Apache
rdk/components/generic/sys_mon_tools/mfr_data	rdk-next	Apache
rdk/components/generic/sys_mon_tools/mfr_utils	rdk-next	Apache
rdk/components/generic/sys_mon_tools/rdklogctrl	rdk-next	Apache
rdk/components/generic/sys_mon_tools/si_cache_parser	rdk-next	Apache
rdk/components/generic/sys_mon_tools/sys_resource	rdk-next	Apache
rdk/components/generic/sys_mon_tools/sys_utils	rdk-next	Apache
rdk/components/generic/sys_mon_tools/udhcpc-opt43	rdk-next	Apache
rdk/components/generic/syslog_helper	rdk-next	Apache
rdk/components/generic/tr69hostif	rdk-next	Apache
rdk/components/generic/trm	rdk-next	Apache
rdk/components/generic/ttsengine	rdk-next	Apache
rdk/components/generic/wifi	rdk-next	Apache
rdk/components/generic/xupnp	rdk-next	Apache
rdk/components/opensource/oe/bitbake	rdk/morty	oe-mirrors
rdk/components/opensource/oe/meta-openembedded	rdk/morty	oe-mirrors

rdk/components/opensource/oe/meta-qt5	rdk/morty	oe-mirrors
rdk/components/opensource/oe/meta-raspberrypi	rdk/morty	oe-mirrors
rdk/components/opensource/oe/meta-virtualization	rdk/morty	oe-mirrors
rdk/devices/intel-x86-pc/emulator/devicesettings	rdk-next	Apache
rdk/devices/intel-x86-pc/emulator/rdkbrowser	rdk-next	Apache
rdk/devices/intel-x86-pc/rdkemulator/gst-plugins-rdk/playersinkbin	rdk-next	LGPL
rdk/devices/intel-x86-pc/rdkemulator/gst-plugins-rdk/qamtunersrc	rdk-next	LGPL
rdk/devices/raspberrypi/devicesettings	rdk-next	Apache
rdk/devices/raspberrypi/gst-plugins-rdk/playersinkbin	rdk-next	LGPL
rdk/devices/raspberrypi/iarmmgrs	rdk-next	Apache
rdk/devices/raspberrypi/wifi	rdk-next	Apache
rdk/tools/tdk	rdk-next	Apache
rdkb/components/generic/CcspLogAgent	rdk-next	Apache
rdkb/components/generic/harvester	rdk-next	Apache
rdkb/components/generic/servicemanager	rdk-next	Apache
rdkb/components/generic/sso	rdk-next	Apache
rdkb/components/generic/startParodus	rdk-next	Apache
rdkb/components/opensource/ccsp/CcspCMAgent	rdk-next	Apache
rdkb/components/opensource/ccsp/CcspCommonLibrary	rdk-next	Apache
rdkb/components/opensource/ccsp/CcspCr	rdk-next	Apache
rdkb/components/opensource/ccsp/CcspDmCli	rdk-next	Apache
rdkb/components/opensource/ccsp/CcspEPONAgent	rdk-next	Apache
rdkb/components/opensource/ccsp/CcspEthAgent	rdk-next	Apache
rdkb/components/opensource/ccsp/CcspHomeSecurity	rdk-next	Apache
rdkb/components/opensource/ccsp/CcspLMLite	rdk-next	Apache
rdkb/components/opensource/ccsp/CcspMoCA	rdk-next	Apache
rdkb/components/opensource/ccsp/CcspMtaAgent	rdk-next	Apache
rdkb/components/opensource/ccsp/CcspPandM	rdk-next	Apache
rdkb/components/opensource/ccsp/CcspTr069Pa	rdk-next	Apache
rdkb/components/opensource/ccsp/CcspWifiAgent	rdk-next	Apache
rdkb/components/opensource/ccsp/CcspXDNS	rdk-next	Apache
rdkb/components/opensource/ccsp/FirmwareSanity	rdk-next	Apache
rdkb/components/opensource/ccsp/GwProvApp	rdk-next	Apache
rdkb/components/opensource/ccsp/GwProvApp-ePON	rdk-next	Apache
rdkb/components/opensource/ccsp/GwProvApp-EthWan	rdk-next	Apache
rdkb/components/opensource/ccsp/hal	rdk-next	Apache
rdkb/components/opensource/ccsp/halinterface	rdk-next	Apache
rdkb/components/opensource/ccsp/hotspot	rdk-next	Apache
rdkb/components/opensource/ccsp/MeshAgent	rdk-next	Apache
rdkb/components/opensource/ccsp/PowerManager	rdk-next	Apache
rdkb/components/opensource/ccsp/sysint	rdk-next	Apache

rdkb/components/opensource/ccsp/TestAndDiagnostic	rdk-next	Apache
rdkb/components/opensource/ccsp/Utopia	rdk-next	Apache
rdkb/components/opensource/ccsp/webui	rdk-next	Apache
rdkb/components/opensource/ccsp/webui-bwg	rdk-next	Apache
rdkb/components/opensource/ccsp/Xconf	rdk-next	Apache
rdkb/devices/intel-x86-pc/emulator/sysint	rdk-next	Apache
rdkb/devices/intel-x86-pc/emulator/tdkb	rdk-next	Apache
rdkb/devices/raspberrypi/hal	rdk-next	Apache
rdkb/devices/raspberrypi/sysint	rdk-next	Apache
rdkb/devices/raspberrypi/tdkb	rdk-next	Apache
rdkb/devices/rdkbemu/ccsp/rdkb	rdk-next	Apache
rdkb/devices/rdkbemu/rdkbemu_xb3	rdk-next	Apache
rdkb/tools/tdkb	rdk-next	Apache
rdkc/components/opensource/configMgr	rdk-next	Apache
rdkc/components/opensource/cvr	rdk-next	Apache
rdkc/components/opensource/httpClients	rdk-next	Apache
rdkc/components/opensource/ledmgr	rdk-next	Apache
rdkc/components/opensource/plugins	rdk-next	Apache
rdkc/components/opensource/rms	rdk-next	Apache
rdkc/devices/raspberrypi/mediastreamer	rdk-next	Apache
rdkc/tools/tdkc	rdk-next	Apache
rdk/components/generic/rdk-oe/meta-cmf	rdk-next	Apache
rdk/components/generic/rdk-oe/meta-cmf-broadband	rdk-next	Apache
rdk/components/generic/rdk-oe/meta-rdk-broadband	rdk-next	Apache
rdk/components/generic/rdk-oe/meta-rdk	rdk-next	Apache
rdk/components/generic/rdk-oe/meta-rdk-ext	rdk-next	Apache
rdk/components/generic/rdk-oe/meta-cmf-raspberrypi	rdk-next	Apache
rdk/components/generic/rdk-oe/meta-cmf-mesh	rdk-next	Apache
rdk/components/generic/rdk-oe/meta-rdk-video	rdk-next	Apache
rdkb/components/opensource/ccsp/RdkCellularManager	rdk-next	Apache