

RMR Testing

The RMR library has both a rich set of unit tests for both coverage and functional verification, and a small set of application based (send/receive) tests for message passing verification. The following paragraphs describe how to create a stand alone test environment, and how to perform the tests.

Container Based Tests

The easiest way to test RMR is to create a Docker image which clones the RMR library from the LinuxFoundation (LF) repo, builds the library, and then can execute the tests. The following Docker file illustrates one way to build the image:

```
FROM ubuntu:16.04
WORKDIR /playpen/gerrit

RUN apt-get update && apt-get install -y gcc git make bash vim cmake g++ ksh

COPY force_update /playpen/
RUN git clone https://gerrit.oran-osc.org/r/ric-plt/lib/rmr
RUN cd /playpen/gerrit/rmr; mkdir .build; cd .build; cmake ..; make install

WORKDIR /playpen/gerrit/rmr

ENV LD_LIBRARY_PATH=/usr/local/lib
CMD [ "ksh" ]
```

The following docker command can be used to build the image (assuming the above is in Dockerfile in the current directory):

```
docker build -t rmr_test:ubuntu .
```

The image is built with the current RMR code from the repo, and to avoid needing to rebuild the image with each RMR change, the test scripts automatically pull and build the latest version before running tests. The following command illustrates how the complete set of build, deb creation, and unit tests can be executed:

```
docker run -it --rm rmr_test:ubuntu ksh test/run_all_test.ksh
```