
BUILD INSTRUCTIONS FOR RVI

This document describes the build process for the RVI project on an Ubuntu 14.01 Linux machine.

Please see `README.md` for a general description of the project and its structure.

Please see `CONFIGURE.md` for details on configuring and launching the system once it has been built.

The first milestone of the RVI project is the HVAC demo. Please see `hvac_demo/README.md` for details on how to setup, launch and drive the demo.

READER ASSUMPTIONS

In order to build the system, the reader is assumed to be able to:

1. Have a basic understanding of Linux system operations.
2. Install packages on the system.

Please note that the configuration process, described in `CONFIGURE.md` may have additional skill requirements.

PREREQUISITES

1. The Ubuntu 14.01 system have the latest updates installed.
2. The user can gain root access to install packages.
3. There is at least 5GB of space available for packages and code.

INSTALLATION PROCESS

INSTALL GIT

Use `apt-get` to install git, which is used to access the Automotive Grade Linux repositories where the code resides:

```
sudo apt-get git
```

INSTALL ERLANG

Install Erlang R16B01, or a later R16 release:

```
sudo apt-get install erlang
```

CLONE THE RVI REPOSITORY

Use the newly installed `git` tool to clone (copy) the RVI repository to the build system.

```
git clone https://gerrit.automotivelinux.org/gerrit/RVI/rvi
```

The clone will be downloaded into a newly created `rvi` subdirectory.

RETRIEVE ADDITIONAL CODE DEPENDENCIES

Move into the newly created `rvi` directory where the code resides.

```
cd rvi
```

Run `make` to pull all necessary repositories into the `deps` subdirectory under the `rvi` directory:

```
make deps
```

The local `rebar` command is used to retrieve the code. See `rebar.config` and `deps/*/rebar.config` for a list of dependencies.

See the [rebar](#) project for a detailed description of the rebar Erlang build tool.

BUILD THE RVI SYSTEM

Run `make` to build the dependency code in `deps` and the top level project in the `rvi` directory.

```
make compile
```

The following warnings are expected, and are not a failure indication:

```
../exo_ssh.erl:18: Warning: undefined callback function code_change/3 (behaviour 'ssh_channel')
...
../bert_challenge.erl:223: Warning: crypto:sha/1 is deprecated and will be removed in in a futu
...
../bert_challenge.erl:230: Warning: crypto:sha/1 is deprecated and will be removed in in a futu
...
../authorize_rpc.erl:31: Warning: function get_certificate_body/2 is unused
```

The compiled code is available under `ebin/` and `deps/*/ebin/`.

CREATE A RELEASE

See `CONFIGURE.md` for details on configuring and creating a developer and production release that can be launched.