Applying Patches

Applying Device-level and SoC-level patches to generic components

Generic RDK code can either use the EXTERNAL SRC model and PATCH model. In the process of integration for patches model, patches to the generic code may be identified and up-streamed to component owners so that these patches can be reviewed and merged into the generic code base. However, there will be a delay between the up-streaming of these patches and the application of these patches to the generic code base. During this time period (i.e. period between up-streaming of patches and merge into generic code base) the builds which are functional on patch model are dependent on these generic patches and may be unable to successfully build or execute. In order to ensure that builds based on patches model continue to build/execute, a mechanism is created that would allow for patches to the generic RDK is made available at the device/soc layers inside a component.

The below steps describe the process used to apply generic patches for SoCs and devices.

1. SoC and device level patches to the generic RDK will be located in full source form inside the component being patches.
2. SoC patches to the generic RDK will be present under the "soc" folder hierarchy inside the component and device-level patches will be present under the "devices" folder hierarchy inside the component.
3. Generic patches will exists as a sparse tree that mirrors the generic folder structure that is being patched.
4. Patches will exist under a "patches" folder inside the "soc" or "devices" folder hierarchy.
5. During environment setup, patches will be exported and will overlay the generic code base.
6. URLs to patches will be present in rdk_env.xml.
7. Patch URLs in rdk_env.xml will be organized on a component-by-component basis and the sequence of URLs will determine the sequence of applying/overlaying patches.
8. Process of upstreaming generic RDK patches will be monitored/tracked as part of the normal patch review process.

Steps to use 'quilt' utility to make patches in Yocto build

Quilt is a software utility for managing "patches" to the source code.

The scenario here is that another team member submitted a change, and you may want to review prior to approving, or you may just need this fix prior to it being merged.

Steps to Modify and Build an Image in a Yocto Workspace without External Source support

1. Find the recipe for the component and add a SRC_URI for the patch file. Add the below line to the recipe of the component you need to modify.

   ```
   SRC_URI += "file://<name>.patch"
   eg: SRC_URI += "file://temp.patch/"
   ```

2. Create a folder named "files" in the same folder of recipe.

3. Create a blank file with the same name given in the recipe.

   ```
   touch files/<name>.patch
   eg: touch files/temp.patch
   ```

4. bitbake <recipe> -c devshell

5. Do "quilt top" and verify if the patch file is reflected there.

   ```
   # quilt top
   patches/temp.patch
   ```

6. quilt add "filename which you need to modify" (If multiple files are there, then add all those)

   ```
   eg: quilt add Makefile.am configure.ac
   ```

7. Make necessary modifications required for these files.

8. quilt refresh

9. Verify if your modifications are reflected in the patch file.

   ```
   vi patches/<name>.patch
   ```

10. Now replace this patch file with original one in files folder.

11. exit
12. Do a Clean and then build.

RDK can either use the external src model and patch model