DL Pager Versioning and Branching Convention

# Introduction

The volume and cadence for DL Pager changes lends its way to using perforce changelist (CL) numbers as natural versioning markers. This eliminates the redundancy of maintaining multiple unique identifiers as well as providing more flexibility with branching

# Internal Convention

The “Main” branch will be marked as latest. From there any CL can be promoted, or branched into an internal branch using the branch point CL as the name for the branch.

Sample Main branch: //source/qcom/qct/core/kernel/dl\_pager\_qurt/main/latest/

If sample main branch contains CLs 1001, 1002, and 1003, the following is an example branch which could be created from 1002

Sample branch point from 1002: //source/qcom/qct/core/kernel/dl\_pager\_qurt/main/1002/

If this branch had CLs 1004, 1006, and 1008, the following would be an example of a branch created from CL in that that branch.

//source/qcom/qct/core/kernel/dl\_pager\_qurt/main/1006/

And so on...

For any given branch, from any given level, a user could use p4v to see the branch history or equivalently use a command like the one below.

P4 filelog –I //source/qcom/qct/core/kernel/dl\_pager\_qurt/main/1006/inc/dlpager\_main.h

# Promotion (External Convention)

Because branching is not utilized at the time of promotion, an alternative method should be used to “pass” along versioning information. This is accomplished through generating release notes along with the promotion automation. If automation is not used, this needs to be created manually and checked in along with the promotion.