

# contrib changes from the AUG side

Antelope Users Group 2015 - San Diego, CA

# Goals

- Support the currently released version of Antelope
- Allow changes to be made to contrib for development versions of Antelope - 5.4post, 5.5pre, etc
- Provide one location for BRTT to build contributed code.
- Install contributed code in `$ANTELOPE/contrib` rather than in the main Antelope tree
- Clarify licensing of contributed code

# Makefile changes

- Antelope contrib Makefiles will now require two lines instead of one:

```
SUBDIR=/contrib  
include $(ANTELOPEMAKE)
```

- `localmake` will need some tweaks
- Makefile changes have been made in a test branch

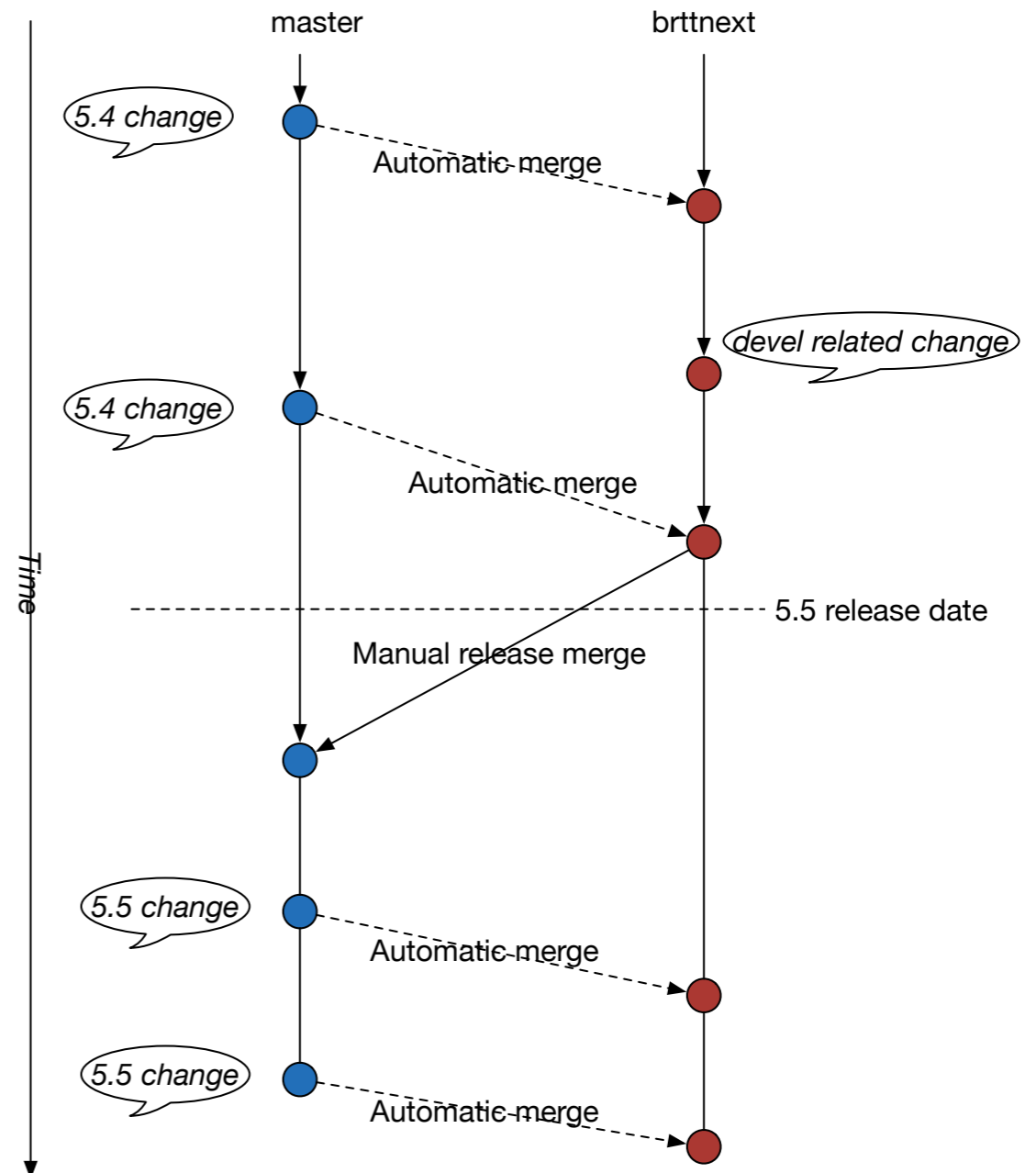
# Makefile changes - implementation

- The default ANTELOPEMAKE rules now look for a file called `$(ANTELOPE)$(SUBDIR)/include/antelopemakerules`
- This evaluates to `/opt/antelope/5.4post/contrib/include/antelopemakerules`
- Source for this file will be part of antelope\_contrib in `first/makes/`
- Relies upon GNU make conventions. Won't work on Solaris make.

# Two branch structure

- master branch remains where most users will commit changes to contributed code
- New branch “brttnext” contains changes for pre-release versions of Antelope that are incompatible with the current release
- Automatic merges of master to brttnext
- BRTT will build and possibly redistribute based on the “brttnext” branch

Contrib master branch with automated merges to brttnext



# Licensing

- GitHub convention is a file called LICENSE at the top level of the module, next to Makefile
- Keeps people from having to scrape  $n$  number of source files for the licensing terms
- You can use what you want, but in order for BRTT to redistribute it, it should be a BSD or MIT license (per Kent's presentation)

# xpls extension

- formerly part of core Antelope, but only used in contrib
- allows perl scripts to be run without setting the full Antelope environment up first
- Useful for non-Antelope cron jobs, network monitoring programs (Nagios/Intermapper), etc

# pod2man Integration

- **P**lain **O**ld **D**ocumentation - from Perl
- Allows you to document the program once, inside the Perl script itself. No need to manually format troff/nroff
- Verbose help has same contents as the man page
- New Macros **PMAN1**, **PMAN3**
- pod2man example: Nagios probes
- (from antelope\_contrib/bin/rt/nagios/nagios\_check\_dlcmdping)  

```
GetOptions("V" => \ $opt_version) ||  
pod2usage(-verbose => 0, -exitval => 1);
```