

Getting Started in the Asterisk Community

Jeff Peeler

Digium Software Developer

September 23, 2008

Presentation Content

- Interacting with the community
- How to submit patches
- Required code format and some good practices
- Helping without writing code
- Helpful resources

Community Interaction

- IRC - real-time Internet chat
 - /server irc.freenode.net
 - main channels: /join #asterisk, #asterisk-dev
 - response time: $t = (0, \infty)$
- Mailing lists
 - <http://www.asterisk.org/support/mailling-lists>
 - main lists: asterisk-users, asterisk-dev

Netiquette

- Common mistakes on the mailing lists
 1. Crossposting
 2. Top posting
 3. Hijacking threads



Patch Submissions

- All patches must be put on bugs.digium.com
- Diffs are to use unified formatting
 - `svn diff > bug8192-1.patch`
 - Make sure patch only has intended changes
- Individual patches should only modify one component or idea
 - `svn diff > fix_mem_leak.patch`
 - `svn diff > fix_whitespace.patch`

Coding Guidelines

- English only please
- No code branding `/* JBP 07/11/2008 */`
 - Large contributions noted in CREDITS file
- C-style comments only (no `//`)
- Attempt to match existing coding style

Coding Guidelines

- Do not declare variables mid-block
- Check input from user
 - Don't use a string with a printf that has formatting
- Utilize the internal API
 - `utils.c / utils.h`, `include/asterisk/*.h`

Code Formatting

- Function calls
 - `ast_indicate(chan, CONTROL_RINGING);`
- Whitespace makes code easier to read
 - `for (str = foo; str; str = str->next) vs`
 - `for(str=foo;str;str=str->next)`
- Braces on same line except for function implementation

Code Formatting

```
switch (foo) {  
  case BAR:  
    blah();  
    break;  
  case FULL_OF_FAIL:  
    handle_failure();  
    break;  
}
```

Code Formatting

```
/* The BAD way - lots of indenting */
```

```
if (foo) {  
    /* .... 50 lines of code ... */  
} else {  
    if (result) /* braces always preferred */  
        result = -1;  
    return result;  
}
```

Code Formatting

```
/* The good way - less indenting required */
```

```
if (!foo) {  
    if (result) {  
        result = -1;  
    }  
    return result;  
}
```

```
/* .... 50 lines of code .... */
```

Variable Naming

Use variable names that make sense

- lower case except when using an external API that uses different case
- Use enum instead of lots of #defines
 - cleaner
 - useful with switches

Doxygen Code Commenting

```
/*!  
 * \brief Do interesting stuff.  
 * \param thing1 interesting parameter 1.  
 * \param thing2 interesting parameter 2.  
 *  
 * This function does some interesting stuff.  
 *  
 * \return zero on success, -1 on error.  
 */  
int ast_interesting_stuff(int thing1, int thing2)  
{
```

Code Development

- Compile with `./configure --enable-dev-mode`
- Create a function or macro for code used more than once
- If a variable is to be reused, save it instead of recomputing

Contributing Without Code

- Report bugs!
- Test patches posted on Mantis
 - `patch -p0 < bug4096-2.patch`
 - report back
- Help others via IRC/Email

Staying Current

- Blogs
 - www.asterisk.org/blog
 - blogs.digium.com
- Twitter users to follow:
 - asteriskpbx
 - digium
 - astricon



Noteworthy URLs

- www.voip-info.org
- www.asteriskdocs.org
- www.asterisk.org
 - www.asterisk.org/developers
- www.digium.com
- snipurl.com/astricon08-community

Noteworthy URLs

- www.voip-info.org
- www.asteriskdocs.org
- www.asterisk.org
 - www.asterisk.org/developers
- www.digium.com
- snipurl.com/astricon08-community

Questions?

