

# New EasyApache: From Here to There and Back Again

Presented by: Daniel Muey



# What's the Same?

- Easy interface to build Apache
- Flexible Curses and WHM Interface

# What's New Outside?

- Builds Apache 2.0 and 2.2 as well as 1.3
- Curses easier to use (Curses::UI mouse support)
- Curses and WHM interfaces parallel
- Custom configuration preserved (sans conflicts)
- Easily build w/ out PHP: faster-safer environment

# What's New Inside?

- Code, Structure/paradigm, Logging
- Goal was:
  - Less and Easy troubleshooting
  - Easy development and integration
  - Quick and simple maintenance



# Less and Easy Troubleshooting

- /usr/local/cpanel/logs/easyapache has all output (STDOUT, STDERR)
- Is the same as verbose output (w/ start/stop timestamps)
- Output formatted & separated visually



# Less and Easy Troubleshooting: The Pros of Conflicts

- Conflict resolution goal:
  - Are not even able to select conflicting opts
  - If forced (EG via a purposeful bad URL or manually edited config, like all PHP 4's):
    - Adds any missing required options and rechecks
    - Removes any conflicting ones and rechecks
    - If it finds missing required options it now know you have circular requirements and stops
    - All output is very descriptively verbose

# Less and Easy Troubleshooting

- Directories:
  - /var/cpanel/easyapache [cPanelEA]
  - /home/cpeasyapache [ /work/cPanelEA/ ]
- Power in Perl's Namespaces:
  - cPanelEA::Apache1::XMod\_Perl =  
cPanelEA/Apache1/XMod\_Perl.pm



# Less and Easy Troubleshooting: What's in a Name?

- 1<sup>st</sup> letter determines when its run
  - X = before Apache, Y = After Apache
- \w+ after 1<sup>st</sup> letter is its name (\_'s become spaces)
- Optional version: \_\_1\_2\_3
- Optional “Note flag”: \_\_S
- YMod\_moto\_\_0\_20\_0\_\_S.pm
- Mod moto (0.20.0) See [projectmoto.org](http://projectmoto.org) for ...



# Less and Easy Troubleshooting

- Look at the source for the source of the problem.
  - Most option modules have very little code
  - Most option modules use the same framework
  - Many errors include the module in question

# Easy Development and Integration

- What changes have been quicker already?
- Anatomy of an “Option Module”:
  - XTMBG.pm is run in XTMBG.pm.build/
  - Optional XTMBG.pm.tar.gz is in .build/

```
package cPanelEA::Apache1::XTMBG;  
our $easyapache_setup = {};  
1;
```



# Easy Development and Integration

- `$easyapache_setup` hashref keys
- ```
{  
  'name' => 'They Might Be Giants',  
  'note' => 'Make a little birdhouse in your apache', #  
    __S.pm  
  'targz' => {},# desc. XTMBG.pm.tar.gz  
  'step' => { # sorted/run by key, 0..9    '0' => {}, #  
    defines a step  
  },
```



# Easy Development and Integration

- \$easyapache\_setup hashref keys
- targz key:
  - 'targz' => {
    - 'url' => 'where it came from',
    - 'md5' => 'md5\_of\_end\_result',
    - 'cd2' => 'working/dir/inside/',
  - },
  - 'targz' => {
    - 'aka' => 'Apache2/XTMBG.pm.tar.gz',
  - },



# Easy Development and Integration

- `$easyapache_setup->{'step'}` keys' values:

```
{ # all CODE refs: my($opt_ht) = @_;  
  'name' => 'configuring',  
  'pre_check' => sub {}, # optional  
  'pre_check_fail' => 1,  
  # undef = continue on regardless,  
  # 0 = silently skip step,  
  # 1 = carp and skip step,  
  # 2 = carp and skip option,  
  # 3 = croak and stop apache,  
  # other = croak
```



# Easy Development and Integration

- `$easyapache_setup->{'step'} keys' val (cont):`  
    `'command' => [qw(echo hello world)],`  
    `# string or array ref of command`  
    `# CODE ref that returns`  
    `# Hint: use a code ref to do some tasks`  
    `#     and return an empty string`  
    `}`
- Others to be documented online later

# Easy Development and Integration

- DSO Example: cPanelEA::Apache2::YMod\_Perl\_\_S
- 'command' key is a CODE ref that returns an empty string

```
sub {
```

```
    my($opt_ref) = @_;
```

```
    # local $opt_ref->{'verb'} = 1;
```

```
    Cpanel::EasyApache::_open3(
```

```
        $opt_ref, "", \@command
```

```
    ); # repeat for each necessary command
```

```
    return "";
```

```
}
```



# Easy Development and Integration

- Source patch Example:
  - cPanelEA::Apache1::XphpSUExec

command CODE ref:

```
my($opt_ref) = @_;
```

```
push @{$opt_ref->{'patches'}}, @patches;
```

Where @patches has path to patch relative to apache's working directory.

# Easy Development and Integration

- “X” Option but needs ap src path example:
  - cPanelEA::Apache1::XMod\_Perl\_\_S

command CODE ref:

```
my $path = Cwd::cwd();
```

```
# path of XMod_Perl__S.pm
```

```
$path =~ s{[.]build/[^/]*/?$}{};
```

```
$opt_ref->{'do_main_in_src'}{$path} =  
'cPanelEA::Apache1::XMod_Perl__S';
```



# Easy Development and Integration

- “X” Option but needs ap src path example:
  - cPanelEA::Apache1::XMod\_Perl\_\_S

Continued:

- do\_main\_src :

```
our $main = sub {  
    my($opt_ref, $ap_build_dir) = @_;  
    _open3.. APACHE_SRC=$ap_build_dir ...  
    $opt_ref->{'Apache1_configure'}  
    .= " --activate-module=src...perl.a";  
}
```



# Easy Development and Integration

- Parents and children, one big happy family:
- Parent is `Apache1/Parent.pm`
  - If you leave out the `$easyapache_setup` hashref, it will be a label instead of a checkbox in the options
- Children are in `Apache1/Parent/Child.pm`
  - follows name space paradigm



# Easy Development and Integration

- OS modules: for those hard to reach places
- Given Apache1/Foo.pm
- If
  - Apache1/Freebsd/Foo.pm exists
  - The OS is freebsd (as per getos())
    - Use Apache1::Foo from Apache1/Freebsd/Foo.pm instead of Apache1/Foo.pm



# Easy Development and Integration

- Do not use option under certain circumstances:
- Simply return false instead of true in your Option module:

```
return 0 if !_widget_is_installed();
```

- Or ensure that widget is installed...

# Easy Development and Integration

- Binary or Library building needs:
  - `Cpanel::EasyApache::_get_path_installed()`
  - Args: `$opt_ref`, `$build_hashref`
  - Ensures its installed and uptodate
  - Returns the path for use in subsequent commands

# Easy Development and Integration

```
my $build_hashref = {  
  'name'      => 'Text Name',  
  'install_path' => '/opt/name/',  
  'itis_up2date' => sub {  
    `/opt/name/bin/name-config -v`  
    =~ m{  
      \s+ 1[.]4[.]0 \s+  
    }xms;  
  }  
  'working_path' => '../mm-1.4.0',
```



# Easy Development and Integration

```
# $build_hashref Continued
'command_list' => [
  [qw(./configure
      --prefix=/opt/name/)],
  ['make'],
  [qw(make test)],
  [qw(make install)],
],
};
```



# Quick and Simple Maintenance

- /scripts/postbuildhttp
  - Email admin,
  - update status database
  - archive log
- Used to be commonly used for DSOs
  - Better to do “Custom modules”

# Quick and Simple Maintenance

- Custom Modules
  - Same as any “Option module”
  - `/var/cpanel/easyapache/cPanelEA/*`
  - Can be for 1.3, 2.0, and/or 2.2
- Useful for distributing your apache mod
- Useful for adding favorite modules
- Useful for provisioning multiple servers



# Quick and Simple Maintenance

- Provisioning ideas:
  - Do apache 1.3, 2.0, 2.2 in config
  - sync conf file & custom “opt mods”
  - Save them at central location and have a script send them to server and easyapache run:
    - `./myprovisionscript 1.2.3.4 tomcat`

# httpd.conf: From 1 to 2 and Back Again

- Current config (vhost, settings, customizations, etc) + blank config built with apache = new config that works with new binary and still has all your data!
- Sanity check when writing
- Version “translation” expandable as new issues are found or new versions of apache are released
- Keep custom configurations when possible



# What's in Store?

- Fully integrate 3<sup>rd</sup> party framework
- Configuration editor:
  - save custom configurations
  - show them as options in the menu



**cPanel**

• mod\_perl, tomcat, mono, no php, php4, php5, etc  
Training Seminar 2006

# Dan's la Fermeture

- Faster upgrades, additions, fixes from cPanel
- **More reliable, intuitive, useful, profitable experience for you!**