

# Cmdr

Andreas Kupries  
ActiveState Software Inc.  
© 2013

20<sup>th</sup> Annual Tcl Conference  
Bourbon Orleans Hotel, New Orleans, LA  
Sep 23 – Sep 27 2013

# Cmdr

- Background
- Help
- Shells
- Architecture
- Future
- Location & References

# Cmdr - Background

- ActiveSupport Project
  - Needed better command line processing for the “stackato” cli application.
  - Tcl's getopt, Tcllib's cmdline
    - Geared towards procedure argument handling
  - Tcllib's TEPAM
    - Existence was forgotten, might have been usable
  - Inspired by ruby gem “mothership”

# Cmdr - Help

- Generation is fully automatic from command declaration.
- 3 standard formats (full, short, list)
- 2 semi-standard formats (“require” to enable)
  - Json → Connectivity to external doc systems
  - SQL → Exploration by query

# Cmdr - Shells

- Based on “linenoise”
  - Steve Bennet's fork of Antirez's original
- Main command line (Actually all officers)
  - Command completion
  - Parameter value completion → Validation type
- Mini shell for individual commands
  - Enable either globally or in command declaration.
  - Parameter name (command) completion
    - Parameter value completion → Validation type
- Interactive entry of single parameters.
  - Must be enabled in the parameter declaration.

# Cmdr – DSL – Outline

```
package require Tcl 8.5
```

```
package require cmdr
```

```
package require foo-backend
```

```
cmdr create ::foo foo {
```

```
    <specification>
```

```
}
```

```
foo do {*}$argv
```

# Cmdr – DSL – Officer

```
officer alias {
```

```
  description {
```

```
    A collection of commands to manage  
    user-specific shortcuts for command  
    entry
```

```
  }
```

```
  <specification>
```

```
}
```

# Cmdr – DSL – Private

```
private add {  
  description {  
    Create a shortcut for a command (prefix).  
  }  
  <specification>  
} ::foo::backend::alias::add
```

# Cmdr – DSL – Parameter I

```
input name {
```

```
    The name of the new shortcut.
```

```
} {
```

```
    validate ::foo::backend::vt::notacommand
```

```
}
```

```
# single value
```

```
# mandatory
```

```
# empty string
```

```
# no notifications
```

# Cmdr – DSL – Parameter II

```
option debug {  
    Activate client internal tracing.  
}  
{  
    undocumented  
    list  
    validate [call@vtype debug]  
    when-complete [lambda {p tags} {  
        foreach t $tags { debug on $t }  
    }]  
}
```

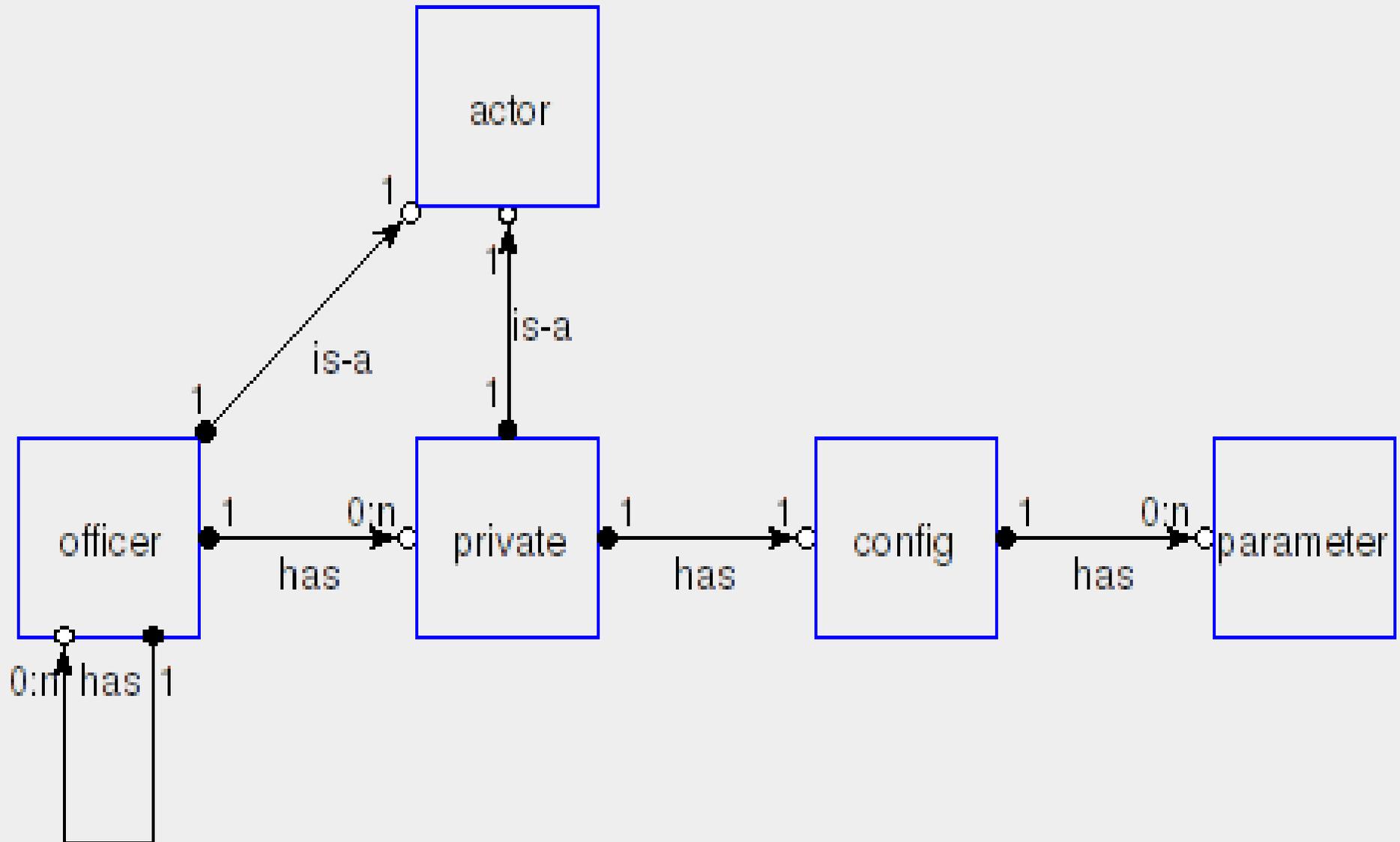
# Cmdr – DSL – Parameter III

```
common .v2 {  
    state checkv2 {  
        Invisible state argument checking that the chosen  
        Target supports the CF v2 API. Use in commands  
        which are v2 only. Note: Requires proper client  
        arguments coming before it.  
    } {  
        immediate  
        generate [call@mgr client isv2cmd]  
    }  
}
```

# Cmdr – DSL – Large Example

- See
  - <https://github.com/ActiveState/stackato-cli/blob/v2/lib/cmd/cmdr.tcl>
- Notifications
  - Actual notifications for deeper layers
  - Cross-parameter checking
  - Command/parameter availability checks
- Custom validation types
- Dynamic extension (aliases)
- ...

# Cmdr - Architecture



# Cmdr - Architecture

- cmdr
  - cmdr::officer → is-a (cmdr::actor)
  - cmdr::private → is-a (cmdr::actor)
    - cmdr::config
      - cmdr::parameter
  - cmdr::help
    - cmdr::help::json
    - cmdr::help::sql
  - cmdr::validate
    - cmdr::validate::common
  - cmdr::util

# Cmdr - Future

- Is the immediate / deferred distinction needed?
- Global options
- Hardwired boolean/presence
  - Can it be generalized?
- Additional and/or generalized validation types
  - For example: Integer ranges.
  - Type-specific interactors
- More help formats? (doctools)
- Overlay of a logical structure on help.

# Cmdr – Location

Where ?

- The Conference eProceedings ( $\equiv$  USB-Stick)
- <http://core.tcl.tk/akupries/cmdr>
- [http://chiselapp.com/user/andreas\\_kupries/repository/cmdr](http://chiselapp.com/user/andreas_kupries/repository/cmdr)

# Cmdr – References

## Supporting packages & tools

- The Conference eProceedings ( $\equiv$  USB-Stick)
- <http://core.tcl.tk/akupries/cmdr>
- <http://core.tcl.tk/akupries/linenoise-utilities>
- <https://github.com/andreas-kupries/tcl-linenoise>
  - <https://github.com/andreas-kupries/linenoise>
  - <https://github.com/msteveeb/linenoise>
    - (<https://github.com/antirez/linenoise>)
- <https://core.tcl.tk/akupries/kettle>