

# Canela Software's Best Practices

Mark Talluto

# Overall Outline

- Programming Style
- 3rd Party Tools
- In-House Tools
- Download Links to Internal Tools

# Programming Style

- Strict compilation mode
- Variable naming
- Comments
- APIs
- App deployment policies
- Server access

# Programming Style

## Strict Compilation Mode

- Turn on 'strict compilation mode'
- Reduces bugs by finding variable spelling errors
- Insures cleaner code - good variable naming
- Best when used with Code Cleaner
- Single before and after & and &&

# Programming Style

## Variable Naming

- Globals: **g**GlobalVarName
- Script Locals: **s**ScriptLocalVarName
- Handler Locals: **t**HandlerLocalVarName
- Constants: **k**ConstantVarName
- Custom Properties: **u**CustomPropName
- Parameters: **p**ParamName, **@r**ParamName (referenced)
- Arrays: tMyArray**A**
- Lists: tMyVar**L** (line delimited) tMyVar**I** (item delimited)
- No single character variables. No one will know what they contain.

# Programming Style

## Comments

- Comment liberally: Make it easy for another dev to work on your code.
- We use `--`, `##`, `//`, `/* */` (Dev choice)
- `-- *****` is used to identify temporary code that should be removed before deployment.
- `-- @@@@` code that still needs work which may be soon or in the distant future.

# Programming Style

## APIs

- When making new handlers, it is best to consider making it an API
- Make your code more approachable/reusable
- Use thoughtful parameter names
- Consider arrays as a parameter when the API demands many inputs. This is useful if you expect the API inputs to evolve.
- Consider in-line params because they work best with auto-complete in script editor

# Programming Style

# Deployment Policies

- Keep historical back-ups of components (allows you to quickly roll back if needed)
- Never push public updates for products out of beta on Fridays (unless you are available to support a bad push)



# Programming Style

## Public/Private Keys

- SSH is your friend...Use it!
- Improves server security by locking access to only certain hardware. You can apply a password to the ssh access for extra security.
- Improves dev performance by allowing you to make calls like `rm -rf` on a directory vs using an FTP program which will be less performant.

# Tools

- LiveCode is our primary IDE and scripting language
- LiveCloud for database and backend when developing with LiveCode
- Appli for rapid layout and development. Provides the best mobile experience.
- GitHub - version control
- BBEdit (Mac), Sublime Text (Mac/Win)
- Navigator (3rd party project browser) - Geoff Canyon
- Flight Recorder (Tracing tool) - Richard Gaskin
- Dropbox, Google Drive, Microsoft Drive
- Alfred (Mac), Ditto (Win): clipboard enhancements
- Microsoft Teams
- ClickUp: project management
- Ansible: server management
- Digital Ocean and Microsoft Azure: data-centers used for LiveCloud backend.
- Canela DevTools: adds quickKeys to LiveCode, code cleaner/linter, orphaned puts, hover info, encrypt/decrypt

# Canela Dev Tools

- Code Cleaner
- Array Browser
- Code Timing with statistical analysis
- Orphaned 'put' finder
- Hover over control details
- Quick keys for many things

# Download Link

- [canelasoftware.com/pub/canela/devtools.zip](https://canelasoftware.com/pub/canela/devtools.zip)
- [canelasoftware.com/pub/canela/bestPractices.pdf](https://canelasoftware.com/pub/canela/bestPractices.pdf)
- [liveCloud.io](https://liveCloud.io)