

COWS

Introduction

COWS Version 1.3

Sean Luke

March 20, 1994

COWS is a simple macro language for NeXTSTEP. It is designed to give you, the user, much more power over an application than you have now. With COWS you will eventually be able to:

- ◆ Control a COWS-compliant application within the application itself through a simple but powerful macro language.
- ◆ Control any COWS-compliant application from *another* COWS-compliant application, even on another machine, using the same macro language. This way, you can stitch many applications together to do custom work for you automatically.
- ◆ Extend the power of a COWS-compliant application by adding COWS function libraries in the macro language or compiled Objective-C.
- ◆ Manipulate any non-COWS-compliant NeXTSTEP application by recording keystrokes and mouse movements, storing these away as COWS functions.
- ◆ Control UNIX shell programs, adding them into the mix of custom stuff you've set up.

If COWS can drum up enough support in the NeXTSTEP community, it could really make a difference, by putting the APIs of applications not only in the

hands of developers, but in the hands of the casual user.

This COWS interpreter package is freely available, though marketing parts of this package as objectware should only be done with the permission of the author. But you are free to make and sell your own interpreter, so long as it is compliant with COWS protocols. We need to develop a community here!

What Comes with COWS

COWS currently consists of five items:

- ◆ The COWS Interpreter, an Objective-C object (and a few associated objects) that can be dropped into any application and hooked up with small effort.
- ◆ The current COWS language specification.
- ◆ A series of protocols describing how COWS and other interpreters communicate with each other, with the application, and with function libraries. This allows more sophisticated (i.e., faster) interpreters, or interpreters with prettier language syntax, to be dropped into an application replacing the COWS interpreter—the application won't know the difference.
- ◆ Objective-C COWS function libraries to extend the power of the COWS macro language. These currently include a standard library, an array library, an inter-process control library, and a demonstration library to show COWS abilities.
- ◆ Demonstration applications into which the COWS interpreter has been dropped.

What's Left to Do

COWS is unfinished. Ultimately, COWS should include:

- ◆ String, math, disk i/o, and system-call libraries.
- ◆ Better COWS IPC (boy do we need *that*).
- ◆ A compiler that converts COWS programs into COWS-compliant Objective-C function libraries (some hacking on the interpreter to make it reentrant will be necessary as well).
- ◆ A library or COWS-compliant app to control UNIX programs.
- ◆ A library (or more likely, a COWS-compliant app) to control non-compliant NeXTSTEP apps through events. Some work may be done with incorporating Puppeteer into the COWS project.
- ◆ A connection kit and palette to make it easier to incorporate COWS into IB.
- ◆ Better error reporting.
- ◆ An alternative language syntax (currently, COWS is LISP-like, but that doesn't have to be its final form).
- ◆ A better, faster, smaller, more robust interpreter.
- ◆ Possibly a client-server interpreter design, though I suspect this may be slower than is acceptable.

How Can I Help?

Please join the fray! COWS needs all the help it can get. There are plenty of function libraries and apps to go around for everyone to work on, and we need test applications, interpreter/language experts, and beta users. You don't have to be a master of COWS to contribute to the programming, though having a good understanding of NeXTSTEP and Obj-C would help. If you have neither, you can still help out in testing the applications, contributing ideas (we programmers always see things with an, um, programmer-point-of-view), and helping evangelize to developers in the marketplace about the need for a macro language design, even if it isn't COWS.

If you're interested in helping out in particular parts of the COWS project, send mail to sean@digaudio.byu.edu.

There are also two e-mail groups and an ftp site associated with COWS:

NeXTInterpreter e-mail group. To discuss macro languages in general and COWS in specific. To join, send mail to sean@alpine.byu.edu.

OpenScript e-mail group. To hack out the specification for a system-wide macro language. This group is open to any comers, including tcl, AppleScript, HyperSense/HyperTalk, Dylan, REXX and A/REXX, and COWS. To join, send mail to openscript@xedoc.com.au.

COWS FTP. There is an anonymous ftp site for COWS and other interpreter-related stuff: [pub/nextinterpreter](ftp://pub/nextinterpreter.alpine.com) at alpine.com.

Thanks to...

- ◆ Glen Henshaw for the COWS name.
- ◆ Benjamin Black for the directory icon.
- ◆ Todd Nathan for the app icon and testing.
- ◆ Don Yacktman, Paul Cardon for testing, support, and ideas.
- ◆ Kris Magnusson and Alpine Computing Microage for the ftp site and mail list drop.
- ◆ Dave Griffiths for Puppeteer.app (which may soon be used), help on Distributed Objects, messaging, etc.
- ◆ Brett Adam for the OpenScript discussion list.