

COWS

Table of Contents

COWS Version 1.3

Sean Luke

March 20, 1994

Applications

- ◆ COWS. An example application to show off the COWS interpreter. COWS contains the Standard, Example, Array, and IPC libraries.
- ◆ COWS2. The identical application (though compiled under the name COWS2) to be used with COWS to show off Inter-Process Communication.
- ◆ *Fred*. A special COWS-compliant version of Fred, an app currently available on archives. Fred contains the Standard and Fred libraries (for a description of the Fred library, see *Fred.rtf*).

Code

- ◆ *COWSInterpreter*. The main COWS interpreter.
- ◆ *COWSProtocols*. Protocols used in COWS communication.
- ◆ *COWS*Node*. A variety of supporting nodes placed in dictionaries in the interpreter or in the interpreter's stack, or passed amongst programs and

libraries. The main data transfer and storage mechanisms in COWS. One exception: The `COWSArrayNode` is used exclusively by the `COWSArrayLibrary`, and by no one else.

- ◆ *COWSStack*. An artificial stack used internally by the interpreter to maintain running programs.
- ◆ *COWSArgumentList*. A subclass of `COWSStack` used to pass arguments to libraries or to the interpreter.
- ◆ *COWSTestBench*. An interface object between the interpreter and the surrounding app. This should be removed if you're placing the interpreter in another app.
- ◆ *COWS*Library*. Interpreter libraries. Of note: the `COWSIPCLibrary` works a bit differently from other libraries. Be careful when adding it to your system. Also, the `COWSExampleLibrary` is only used to demonstrate COWS using the app's Jungle Gym. If you're putting COWS in a real app, don't include this library.

To test inter-app communication, you can build two different versions of COWS (`COWS` and `COWS2`) which have different internal names, and so are posted as different apps on the Network Name Server. To build an app with any name, just change the *Project Name* in the *Attributes* section of the project, save the project, and recompile.

Documents

- ◆ *COWS Introduction*. An introduction to the COWS system. For developers and users.
- ◆ *Fred*. A short description the new improved (and slower) Fred.
- ◆ *COWS Language Concepts*. An introduction to the COWS language. Includes a description of basic data types, special forms, and the COWS Standard Function Library. For users.

- ◆ *COWS Jungle Gym Functions.* COWS function descriptions to control the app's Jungle Gym through the COWS Example Library. For users.
 - ◆ *COWS Array Library.* Descriptions of how to do arrays with COWS. For users.
 - ◆ *COWS IPC Library.* COWS inter-process functions to try out. Make sure you've got two *different* COWS-compliant apps running (two are supplied). For users.
 - ◆ *COWS Language Formal Specification.* Formal description of the language. For those computer scientists amongst us.
 - ◆ *Test Code.* COWS functions to try out. For users.
-
- ◆ *COWS Interpreter Tour.* An introduction to how to set up and use the COWS Interpreter. For developers.
 - ◆ *COWS Interpreter Object Hierarchy.* Graphical pictures of the basic COWS object relationships. For developers.
 - ◆ *COWS Palette.* Used to make the COWS Interpreter Object Hierarchy diagrams. Use this as you like, when submitting libraries or modifications to the interpreter, etc.. For developers.
 - ◆ *COWS Parser Grammar.* How the COWS Interpreter breaks code into useable chunks before interpreting. For really-into-COWS developers.
 - ◆ *COWS Interpreter Driving System.* A description of some of the bizarre private COWS Interpreter methods. For *excessively*-into-COWS developers.
-
- ◆ *COWS Protocols.* A description of protocols used in COWS communication. For developers.
 - ◆ *COWS Bugs.* Known bugs and problems with version 1.3 of COWS.