# Autodoc

Tim Daly

March 24, 2013

**Abstract**

We want to generate the reference information from axiom source files automatically.

# Contents

By design the chunk name has 3 fields. The first field is one of {category, domain, package}. The second field is the abbreviation. The third field is the full name. This routine will return two lists, the first is the list of abbreviations and the second is the list of full names.

We keep looping until we run out of input, at which point we throw out of the loop and catch it here.

⟨*until done*⟩≡

```
 (catch 'done
```

We open the source file

⟨*open source file*⟩≡

```
  (with-open-file (in sourcefile)
```

We read each line, trimming off trailing spaces

⟨*for each line do*⟩≡

```
   (loop
    (setq expr (read-line in nil 'done))
    (when (eq expr 'done) (throw 'done nil))
    (setq expr (string-trim '(#\space) expr))
```

We look for lines that begin with a chunk name that starts with package, domain, or category and end with an equal sign (which indicates a chunk definition).

⟨*when package domain or category definition*⟩≡

```
    (when
     (and (> (length expr) 4)
          (or
           (string= "<<pa" (subseq expr 0 4))
           (string= "<<do" (subseq expr 0 4))
           (string= "<<ca" (subseq expr 0 4)))
          (char= (elt expr (1- (length expr))) #\=))
```

We remove the 3 characters, two > and an = which make up the chunk name.

⟨*remove the trailing chunk characters*⟩≡

```
     (setq expr (subseq expr 0 (- (length expr) 3)))
```

We get the third field from the string which is the long name.

⟨*capture the long name*⟩≡

```
     (setq point (position #\space expr :from-end t :test #'char=))
```

We get the second field from the string which is the abbreviation.

⟨*capture the abbreviation*⟩≡

```
     (setq mark
      (position #\space
       (string-right-trim '(#\space)
        (subseq expr 0 (1- point))) :from-end t))
     (push (string-trim '(#\space) (subseq expr mark point)) names)))))
```

⟨*srcabbrevs*⟩≡

```
(defun srcabbrevs (sourcefile)
 (let (in expr start end names longnames)
⟨until done⟩
⟨open source file⟩
⟨for each line do⟩
⟨when package domain or category definition⟩
⟨remove the trailing chunk characters⟩
⟨capture the long name⟩
      (format t "~a ~a ~a~%" expr point longnames)
⟨capture the abbreviation⟩
  (values names longnames))))
```

# 1 Doit

⟨*doit*⟩≡

```
(srcabbrevs "/new/patch38/src/algebra/acplot.spad.pamphlet")
```

# 2 Autodoc

⟨*autodoc*⟩≡

```
⟨srcabbrevs⟩
⟨doit⟩
```

# 3 Makefile

⟨*⟩≡

```
TANGLE=/usr/local/bin/notangle -t8
WEAVE=/usr/local/bin/noweave -delay
LISP=/new/patch38/obj/linux/bin/lisp

all: code doc run

code: autodoc.lisp.pamphlet
	@${TANGLE} -Rautodoc autodoc.lisp.pamphlet >autodoc.lisp

doc: autodoc.lisp.pamphlet
	@${WEAVE} autodoc.lisp.pamphlet >autodoc.tex
	@latex autodoc.tex
	@latex autodoc.tex

run: autodoc.lisp
	@cat autodoc.lisp | ${LISP}

remake:
	@${TANGLE} autodoc.lisp.pamphlet >Makefile
```

# References

[1] nothing