NAUTILUS

20,000 Leagues under the tCl

Lindsay F. Marshall

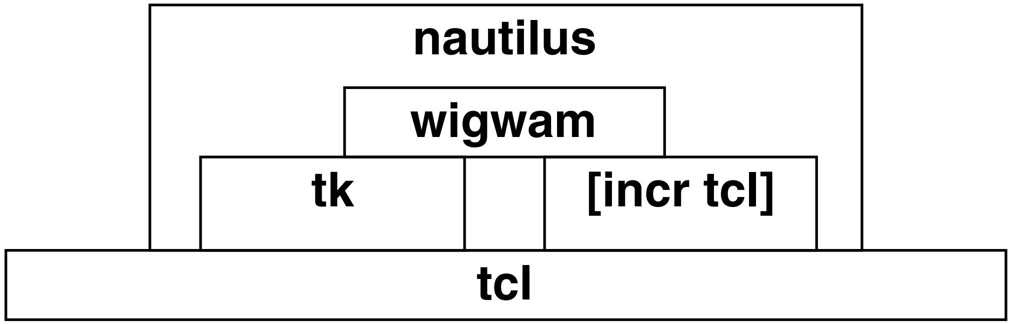
Department of Computing Science
University of Newcastle upon Tyne

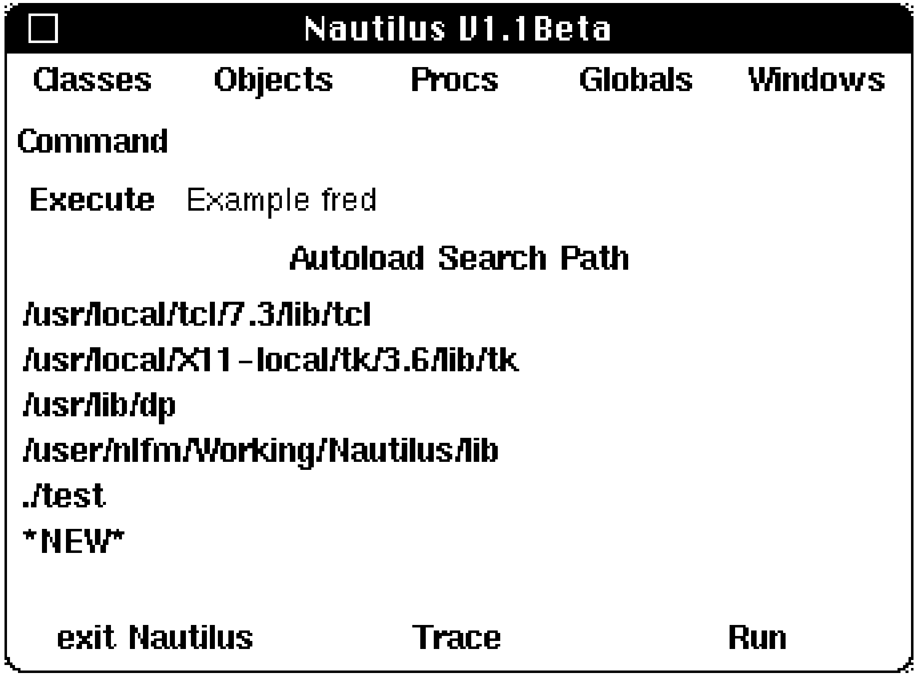
Lindsay.Marshall@newcastle.ac.uk

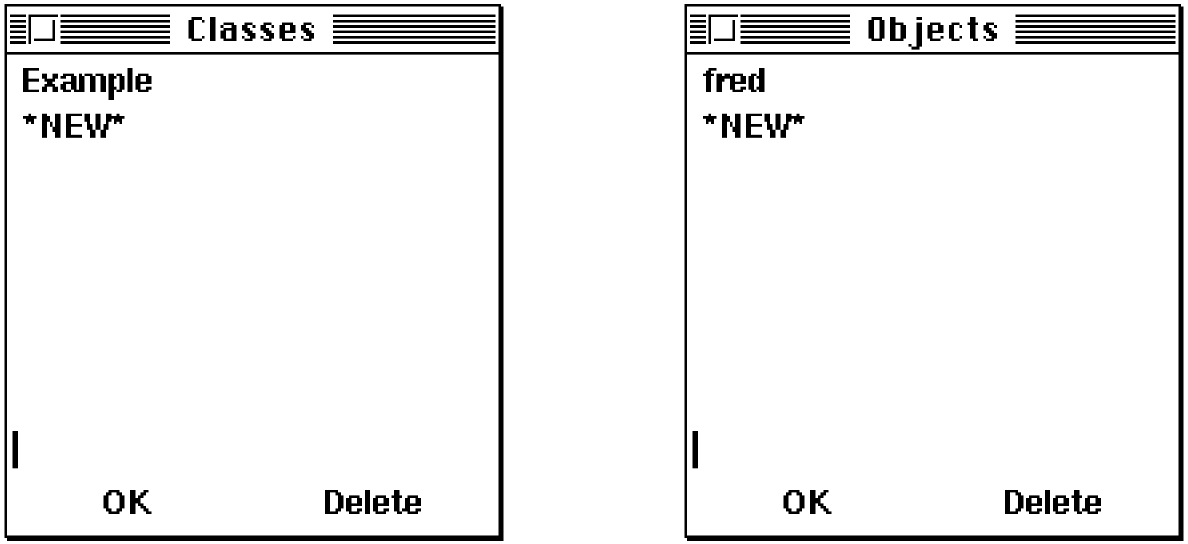
Aims

- Class Browser
- Debugger
- All written in [incr tcl]

Structure




Nautilus V1.1Beta
Classes
Objects
Procs
Globals
Windows
Command
Execute
Example fred
Autoload Search Path
/usr/local/tcl/7.3/lib/tcl
/usr/local/X11-local/tk/3.6/lib/tk
/usr/lib/dp
/user/nlfm/Working/Nautilus/lib
./test
NEW
exit Nautilus
Trace
Run


Classes
Example
NEW
OK
Delete
Objects
fred
NEW
OK
Delete

Procs

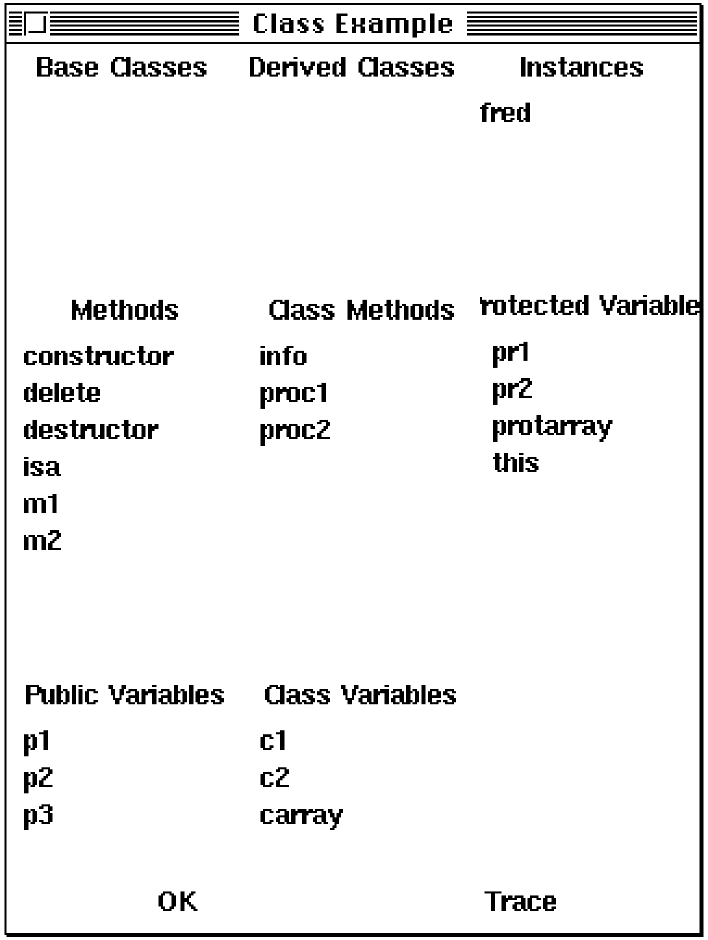
Acl+
Acl-
AclCheck
auto_execok
auto_load
auto_load_all
auto_mkindex
auto_reset
close
destroy
dp_AcceptRPCConnection
dp_AppendTrigger
dp_AppendTriggerUnique
dp_CheckHost
dp_CleanupRPC
dp_ClearTriggers

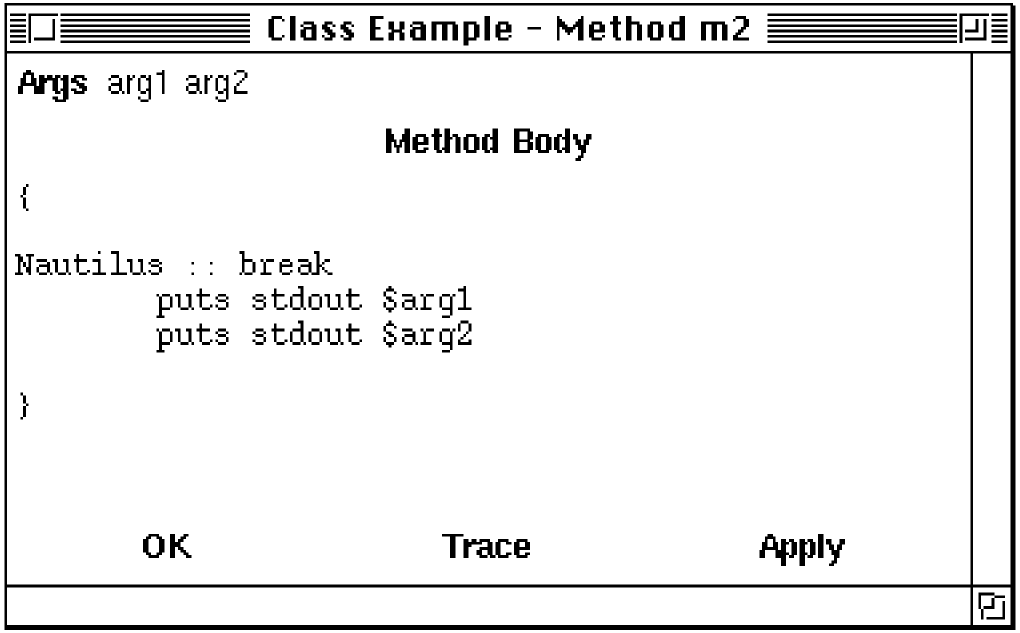
OK Delete

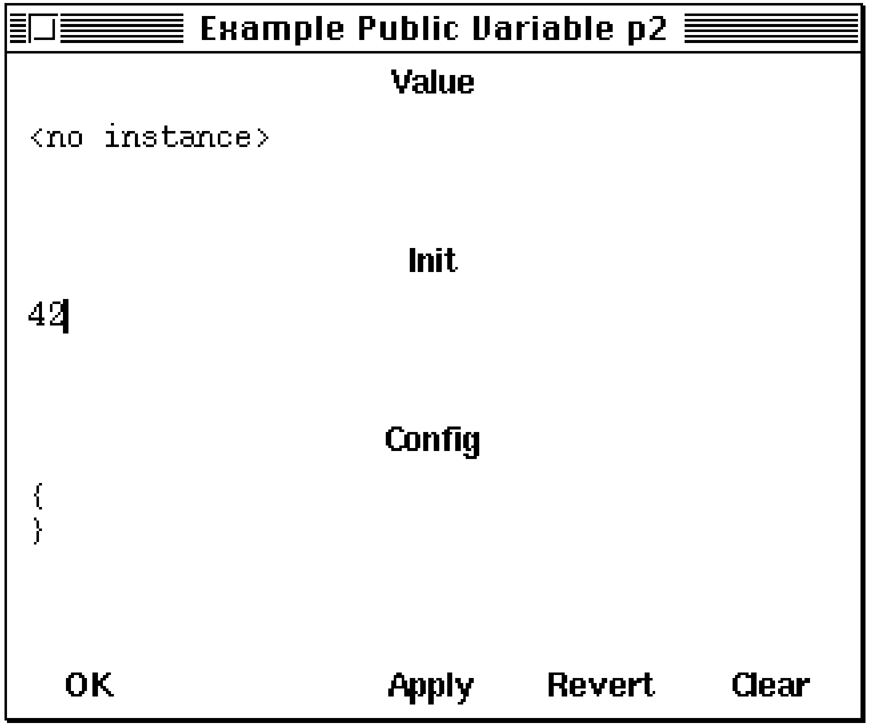
Globals

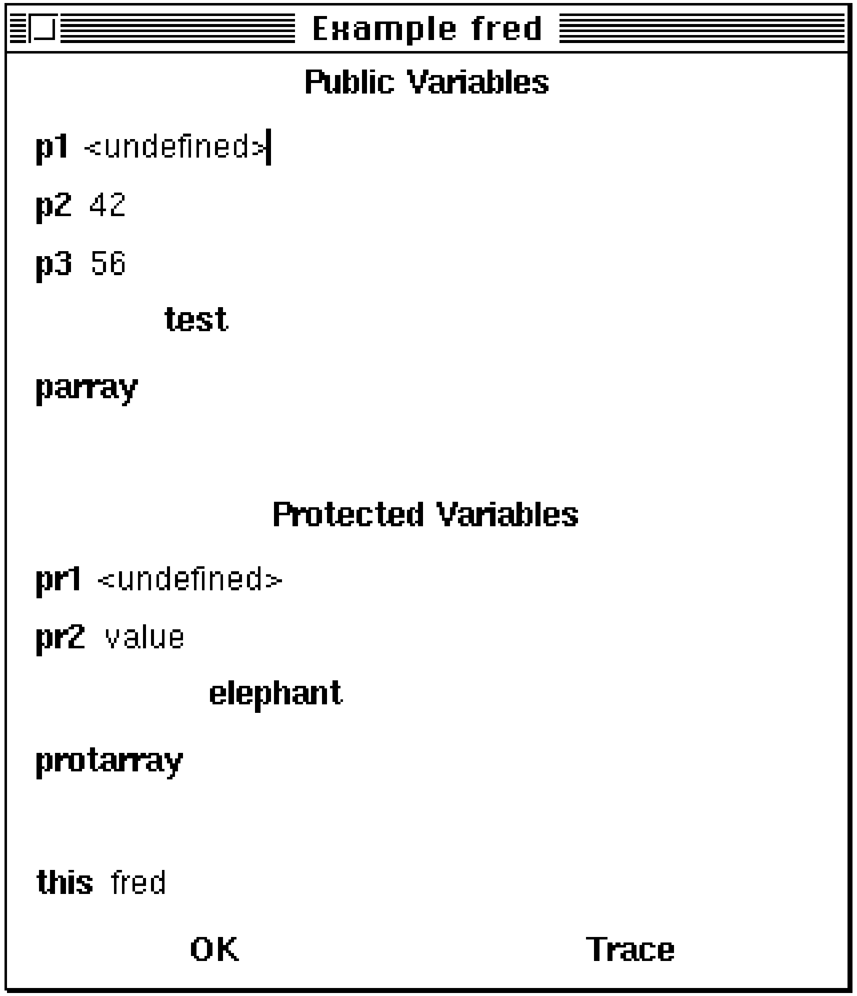
_objects
argc
argv
argv0
auto_index
auto_oldpath
auto_path
dp_Acl
dp_RPROCtable
dp_library
dp_version
env
errorCode
errorInfo
name
tcl_interactive

OK Delete


Class Example
Base Classes
Derived Classes
Instances
fred
Methods
constructor
delete
destructor
isa
m1
m2
Class Methods
info
proc1
proc2
rotected Variable
pr1
pr2
protarray
this
Public Variables
p1
p2
p3
Class Variables
c1
c2
carray
OK
Trace


Class Example - Method m2
Args arg1 arg2
Method Body
{
Nautilus :: break
puts stdout $arg1
puts stdout $arg2
}
OK
Trace
Apply

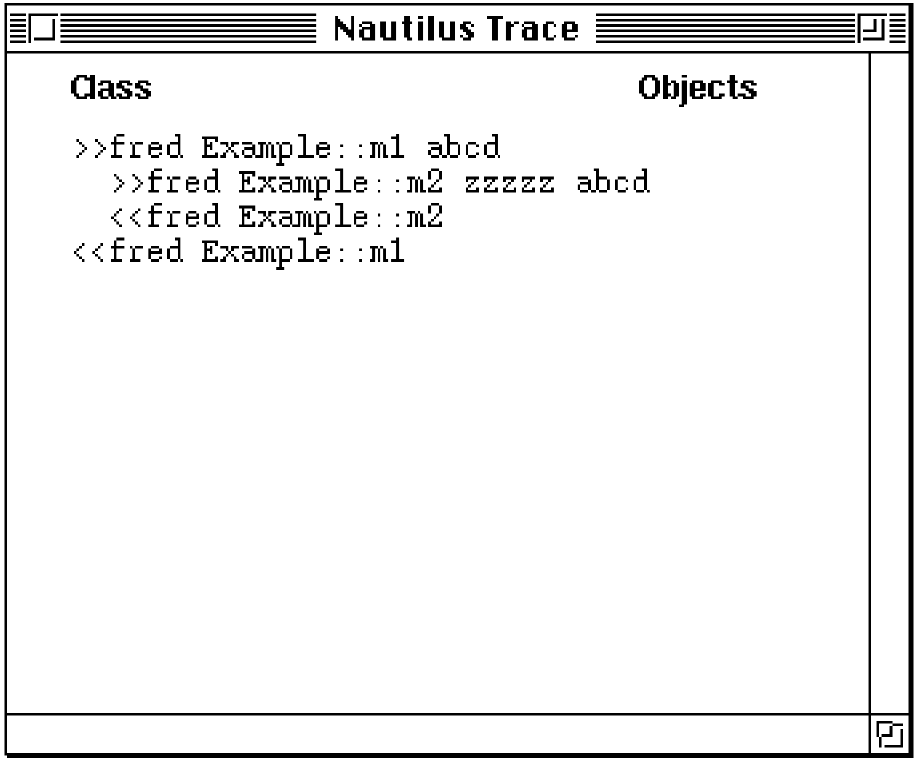

Example Public Variable p2
Value
<no instance>
Init
42
Config
{
}
OK
Apply
Revert
Clear


Example fred
Public Variables
p1 <undefined>
p2 42
p3 56
test
parray
Protected Variables
pr1 <undefined>
pr2 value
elephant
protarray
this fred
OK
Trace

Breakpoint

Example m1 abcd
Example m2 zzzzz abcd
Nautilus :: break

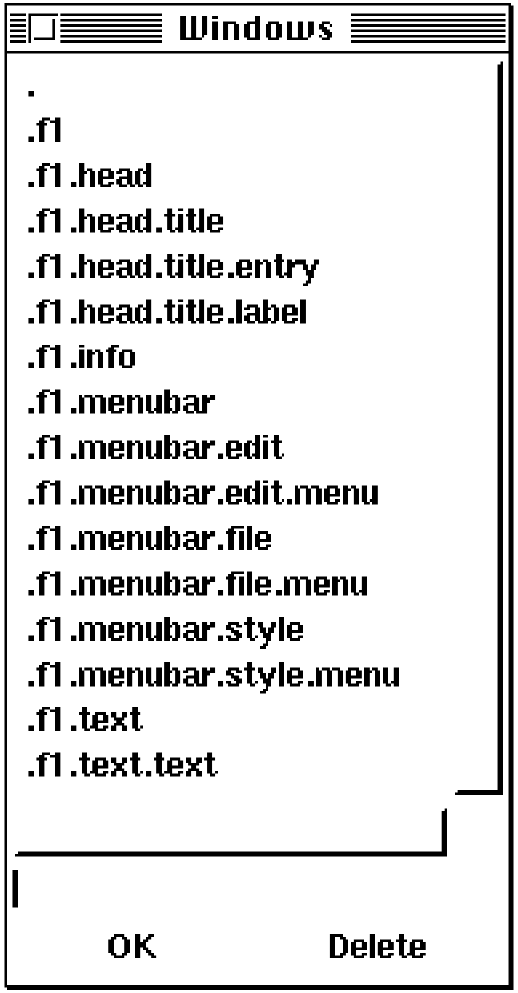
OK

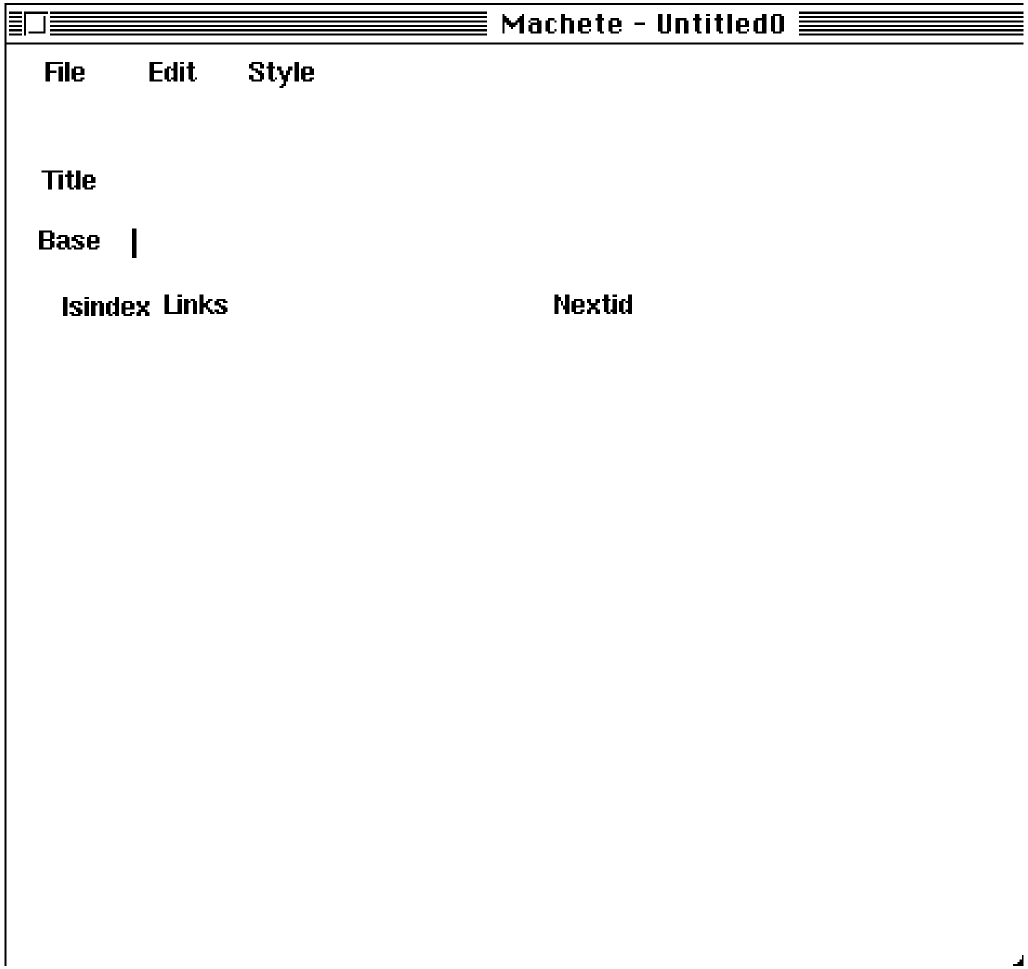

Nautilus Trace
Class
Objects
>>fred Example::m1 abcd
>>fred Example::m2 zzzzz abcd
<<fred Example::m2
<<fred Example::m1

Nautilus Trace

Class **Objects** **Procs**

```
   >>File0 File::constructor {-file {}}
     >>.f1 Machete::constructor {}
       >>.f1 Window::constructor {-title {Machete - Untitled0} -iconname Untitled0 -mi
nsize {10 1}}
            >>.f1 Window::title config {Machete - Untitled0}
            <<.f1 Window::title config
            >>.f1 Window::iconname config Untitled0
            <<.f1 Window::iconname config
            >>.f1 Window::minsize config {10 1}
            <<.f1 Window::minsize config
         >>.f1 Window::configure {-title {Machete - Untitled0} -iconname Untitled0 -mi
nsize {10 1}}
         <<.f1 Window::configure
       <<.f1 Window::constructor
     <<.f1 Machete::constructor
   <<File0 File::constructor .f1
```


Windows
.
.f1
.f1.head
.f1.head.title
.f1.head.title.entry
.f1.head.title.label
.f1.info
.f1.menubar
.f1.menubar.edit
.f1.menubar.edit.menu
.f1.menubar.file
.f1.menubar.file.menu
.f1.menubar.style
.f1.menubar.style.menu
.f1.text
.f1.text.text
OK
Delete


Machete - Untitled0
File
Edit
Style
Title
Base
Isindex
Links
Nextid

Nautilus Trace

Class **Objects** **Procs**

```
    }
    if {!0} {
    }
  set machete(lib) .
  lappend auto_path ./lib
    foreach c {} {
    }
  lappend auto_path /usr/local/X11-local/wigwam/current/lib
    foreach c {} {
    }
  set machete(window) 0
  set machete(untitled) 0
  set machete(names) 0
  set machete(styles) {h1 h2 h3 h4 h5 h6 a em strong code samp kbd var dfn
 cite address tt b i u}
 cite address tt b i u}
  set rendering(TT) -adobe-courier-medium-r-normal--10-100-75-75-m-60-iso8859-1
  set rendering(ADDRESS) {-adobe-new century schoolbook-medium-i-normal--10-100-75-75
-p-60-iso8859-1}
  set rendering(A) -b&h-lucida-medium-i-normal-sans-10-100-75-75-p-59-iso8859-1
  set rendering(BLOCKQUOTE) {-adobe-new century schoolbook-medium-i-normal--10-100-75
-75-p-60-iso8859-1}
  set rendering(H1) -adobe-helvetica-bold-r-normal--18-180-75-75-p-103-iso8859-1
```

How?

```
itcl_class Example {
    #
    constructor {config} {
        set carray(1) 42
        set parray(test) "Lindsay Marshall"
        set protarray(elephant) {a b c d}
    }
    #
    destructor { }
```

```
itcl_class Example {
  constructor {args} {
    global _nts_body
    eval $_nts_body(Example,init)

    _nts_put _nts_cfg [eval _nts__nts_config $args]
    Nautilus :: tracein \
      $this Example constructor  $args
    _nts_put val [_nts_constructor  $_nts_cfg]
    Nautilus :: traceout \
      $this Example constructor $val
  }
  method _nts_constructor args {
    global _nts_body
    _nts_put config $args
    return [eval \
       $_nts_body(Example,method,constructor)]
  }
```

```
destructor {
   Nautilus :: tracein $this Example destructor
   _nts_put val [_nts_destructor]
   Nautilus :: traceout $this Example destructor
}

method _nts_destructor {} {
   global _nts_body
   return [eval \
      $_nts_body(Example,method,destructor)]
}
```

```
method m1 {arg1} {
    set l1 val1
    set l2 val2
    set la(45) 45
    if {$l1 == $l2} { puts then } { puts else }
    m2 zzzzz $arg1
}
#
method m2 {arg1 arg2} {
    Nautilus :: break
    puts stdout $arg1
    puts stdout $arg2
}
```

```
method m1 arg1 {
    Nautilus :: tracein $this Example m1  $arg1
    _nts_put val [_nts_m1  $arg1]
    Nautilus :: traceout $this Example m1 $val
}

method _nts_m1 arg1 {
    global _nts_body
    return [eval $_nts_body(Example,method,m1)]
}
```

```
proc proc1 {arg1} {
     proc2 xxxx $arg1
}
#
proc proc2 {arg1 arg2} {
     Nautilus :: break inside proc2
}
```

```
proc proc1 arg1 {
   Nautilus :: tracein $this Example proc1  $arg1
   _nts_put val [_nts_proc1  $arg1]
   Nautilus :: traceout \$this Example proc1 $val
}

proc _nts_proc1 arg1 {
   global _nts_body
   return [eval $_nts_body(Example,proc,proc1)]
}
```

```
        common c1
        common c2 "c2"
        common carray
        #
        public p1
        public p2 42
        public p3 56 {puts stdout $p3}
        public parray
        #
        protected pr1
        protected pr2 "value"
        protected protarray
}
```

```
public p3 {} {
   Nautilus :: tracein $this Example \
       {p3 config} $p3
   _nts_put val [_nts__nts_config_p3]
   Nautilus :: traceout $this Example \
       {p3 config} $val
}

method _nts__nts_config_p3 {} {
   global _nts_body
   return [eval $_nts_body(Example,public,p3)]
}
```

```
method _nts__nts_config config { return $config }

method _nts__nts_mdo args {
    return [eval [join $args]
}

proc _nts__nts_pdo args {
    return [eval [join $args]]
}
```

Gnarly bits

- Line by line Tracing
- Bindings
- Mouse interactions
- Quoting

What else?

- Click to add breakpoints
- Interface to variable tracing
- Value tracing (for checking quoting!)
- Syntax directed editing commands
- Better trace control
- Better interface design
- …