

# EXHIBIT F

**JDX 272**

**PA-NAT-00000113**

USDC-EDTX  
6:09-CV-446 LED

**JDX 272**

```
1 <!DOCTYPE hhtml SYSTEM>
2 <HHTML>
3 <TITLE>ViolaWWW Hypermedia Browser</TITLE>
4 <SECTION>
5 <VOBJF>violaWWWTitle.v</VOBJF>
6 <P>
7 ViolaWWW implements an XWindows browser to the
8 <A HREF="http://info.cern.ch/hypertext/WWW/TheProject.html">World Wide Web</A>
9 system.
10 <P>
11 This browser has the
12 <A HREF="help.hhtml">
13 basic features</A> of of hypertext navigation mechanisms: clickable
14 hot-words, history list. It has keyword search entry field, and can
15 do page cloning (multiple windows). There exists a simple printer
16 hook, and a document source viewer.
17 <P>
18 This version of ViolaWWW support two separate DTDs -- HTML (which
19 includes extensions such as inlined images) and
20 <A HREF=".../docs/viola/aboutHHTML.hhtml">
21 HHTML</A>. A simple
22 <A HREF=".../docs/viola/aboutStylesheet.hhtml">
23 stylesheet mechanism</A>
24 is supported. The book-mark lists have been depreciated.
25 <P>
26 ViolaWWW is built using the
27 <A HREF=".../docs/viola/brief.hhtml">
28 Viola hypermedia toolkit</A>, and now also comes with
29 <A HREF=".../docs/viola/aboutMotifFront.hhtml">
30 a Motif front end</A>.
31 <P>
32 <EMPH>See the <A HREF=".../docs/viola/violaWWWdemo.hhtmlb">
33 demo documents</A></EMPH>.
34 <P>
35 Connect to the <A HREF="http://petra.berkeley.edu/ora.com/viola/central.html">
36 Viola Central</A> for more info regarding this software.
37 </SECTION>
38
39 <SECTION>
40 <HR>
41 <H2>Acknowledgements</H2>
42 <P>
43 The <A HREF="http://xcf.berkeley.edu/ht/xcf.html">
44 Experimental Computing Facility</A> (XCF) at U.C. Berkeley supported
45 the initial work on the Viola project. Current work on Viola and its
46 applications is supported by <A HREF="http://ora.com/">
47 O'Reilly & Associates</A>.
48 <P>
49 Most importantly, my sincere thanks go to all the
50 <A HREF="http://petra.berkeley.ora.com/viola/www/doc/credits.hhtml">
51 people who've contributed</A> to Viola in various ways -- with bug reports,
52 suggestions, code patches, moral encouragements, etc. You know who you are.
53 Thanks.
54 <HR>
55 <H2>Contacts</H2>
56 <P>
57 Please address to me the ViolaWWW browser specific comments and questions.
58 There is also a <CMD>www-talk@info.cern.ch</CMD> mailing list.
59 and a <CMD><A HREF="news:comp.infosystems.www">comp.infosystems.www
60 </A></CMD> news group.
61
62 <ADDRESS>
63 <P><A HREF="http://xcf.berkeley.edu/ht/people/wei/disclaimer.html">
64 Pei Y. Wei</A>
65 <P>Developer, O'Reilly & Associates, Digital Media Group
66 <P><CMD>pei@ora.com</CMD>
67 </ADDRESS>
68 </SECTION>
69 </HHTML>
```

```
1 <!DOCTYPE html SYSTEM "html.dtd">
2 <HTML>
3 <HEAD>
4 <TITLE>Test misc</TITLE>
5 </HEAD>
6 <BODY>
7 <H1>Header 1</H1>
8 <H2>Header 2</H2>
9 <H3>Header 3</H3>
10 See the <A HREF=""/>etc/motd</A> for useless messages.
11 Here is an ordered list:
12 <OL>
13 <LI>socks
14 <LI>scarf
15 <LI>pillows
16 </OL>
17 And an unordered list.
18 <UL>
19 <LI>rubber ball.
20 <LI>I recollect that, when a stripling, my first exploit in
21 squirrel-shooting was in a grove of tall walnut-trees that shades
22 one side of the valley. I had wandered into it at noontime, when
23 all nature is peculiarly quiet, and was startled by the roar of
24 my own gun, as it broke the Sabbath stillness around and was
25 prolonged and reverberated by the angry echoes.
26 <LI>If ever I should
27 wish for a retreat whither I might steal from the world and its
28 distractions, and dream quietly away the remnant of a troubled
29 life, I know of none more promising than this little valley.
30 </UL>
31 Definition lists:
32 <DL>
33 <DT>First term
34 <DD>Definition paragraph starts here.
35 <DT>Second term
36 <DD>Second definition paragraph starts here.
37 </DL>
38 Here is a dir (x2):
39 <DIR>
40 <LI>socks 0
41 <LI>scarf 1
42 <LI>pillows 2
43 <LI>apple 3
44 <LI>kiwi 4
45 <LI>banana 5
46 <LI>pinball 6
47 </DIR>
48 <ADDRESS>Angel666
49 <P><KBD>angel666alpha.div</KBD>
50 </ADDRESS>
51 foo
52 </BODY>
53 </HTML>
```

```

1 <!DOCTYPE hmml SYSTEM>
2 <HMML>
3 <TITLE>test input</TITLE>
4
5 <H1>FORM samples</H1>
6 <P>This document contains a variety of forms as supported by viola, in HMML.
7
8 <FORM ACTION="mailto:pei@ora.com">
9 <MR>Subject: A survey</MR>
10 <P>Please enter your name:
11 <INPUT name="UserName" envvar="USER">
12 <P>age:
13 <INPUT name="Age" type="percent" value=10 envvar="AGE">
14 <P>
15 Sex:
16 <HPANE>
17 <INPUT maxwidth=30 name="sex" value="male" type="radio"><P>Male
18 <INPUT maxwidth=30 name="sex" value="female" type="radio"><P>Female
19 </HPANE>
20 </FORM>
21
22 <FORM ACTION="mailto:pei@ora.com">
23 <H2>Pizza sizes</H2>
24 <HPANE>
25 <INPUT maxwidth=30 name="size" value="small" type="radio" checked><P>small
26 <INPUT maxwidth=30 name="size" value="medium" type="radio"><P>medium
27 <INPUT maxwidth=30 name="size" value="large" type="radio"><P>large
28 </HPANE>
29
30 <HR>
31 <H2>Toppings</H2>
32 <P>add 25 cents for each topping
33 <HPANE>
34 <INPUT maxwidth=30 name="Anchovies" type="checkbox"><P>Anchovies
35 <INPUT maxwidth=30 name="Salam" type="checkbox"><P>Salam
36 <INPUT maxwidth=30 name="Sausage" type="checkbox"><P>Sausage
37 </HPANE>
38
39 <HPANE>
40 <INPUT maxwidth=30 name="GreenPeppers" type="checkbox" checked><P>Green peppers
41 <INPUT maxwidth=30 name="Mushrooms" type="checkbox" checked><P>Mushrooms
42 <INPUT maxwidth=30 name="Artichokes" type="checkbox" envvar="TOPPING_ARTICHOKES"><P>Artichokes
43 </HPANE>
44
45 <HR>
46 <P>Red hot pepper density: (number of shakes -- 9 is ideal)
47 <INPUT name="HotPepper" type="percent" value=9>
48 <P>Cheese density: (50% is recommended)
49 <INPUT name="Cheese" type="percent" value=50>
50 <P>Tomato sauce depth: (in millimeters. 3 is nice and moist)
51 <INPUT name="TomatoSauce" type="percent" value=3>
52
53 <HR>
54 <H2>Option menu</H2>
55 <HPANE>
56 <OPTIONS name="cities" maxwidth=150>
57 <OPTI value="berlin">Berlin</OPTI>
58 <OPTI value="cairo">Cairo</OPTI>
59 <OPTI value="sf" checked>San Francisco</OPTI>
60 </OPTIONS>
61 <P>Which city are you going to?
62 aftereffect afterglow afterimage afterlife aftermath afternoon
63 afterthought afterward afterward again against Agamemnon agate Agatha
64 agave age Agee agenda agent agglomerate agglutinate agglutinin aggravate
65 aggregate aggression aggressive aggressor aggrieve aghast
66 </HPANE>
67
68 </FORM>
69
70 <P>Radio buttons.
71 <FORM ACTION="mailto:pei@ora.com">
72 <H2>Signature</H2>
73 <INPUT name="sig" type="doodle">
74 </FORM>

```

```

75
76 </HMML>

```

```
1 <!DOCTYPE hhtml SYSTEM>
2 <HHTML>
3 <TITLE>RandomMyster's Waves</TITLE>
4 <DL>
5 <DT>Title<DD>Brush Motion
6 <DT>Artist<DD>Rando "random art does not apologize" Myster
7 <DT>Date<DD>1993, March 7, before breakfast
8 </DL>
9 <VOBJF>colorwavy.v</VOBJF>
10 <P>
11 Warning: playing with color too much will take all the color resources...
12 (Yes, this is a bug).
13 </HHTML>
```

```
1 <!DOCTYPE hmm1 SYSTEM>
2 <HMM1>
3 <TITLE>Embedded Viola Objects</TITLE>
4 <HI>Embedding Viola Objects in HMM1 Document</HI>
5 <P>
6 The <CMD>&lt;VOBJF&gt;</CMD> tag can be used to insert viola applications
7 into a document. Since a viola objects are programmed, this is also
8 a way to link programming into the document.
9 <P>
10 The <CMD>&lt;VINSER&gt;</CMD> tag has similar functionality, but executes
11 the script and inserts the result of the execution.
12 <P>
13 Of course, too much violaism reduces the portability of your document
14 on the World Wide Web. But anyway, this is neat... This tag should be
15 deprecated and use <CMD>&lt;EMBED&gt;</CMD> instead, and treat viola
16 objects as foreign data types, like postscript.
17 <P>
18 Ideas for dealing with security include: disabling these objects'
19 privilege to execute system sensitive commands such as <CMD>system(</CMD>;
20 Limit the scope of the messaging to be only among the objects related
21 to the document... This is an open question, so I (pei@ora.com) would
22 be interested in ideas.
23 <P>
24 Here are a couple of examples of embedded viola applications (objects).
25 <HR>
26 <H2>User Survey</H2>
27 <P>This form isn't done with HTML+ input form. It's a viola application
28 that's a form. This particular example is obsolete since the advent of
29 HTML+ forms, but potentially this is useful for building arbitrary new
30 form devices that aren't in the HTML+ standard.
31 <VOBJF>.../demo/objs/vwq.v</VOBJF>
32 <HR>
33 <H2>Wave motion</H2>
34 <P>More useful would be some kind of interactive equation plotter.
35 <VOBJF>.../demo/objs/wave.v</VOBJF>
36 </HMM1>
```

```

1 \name {doodle}
2 \class {vpane}
3 \children {doodle.ctr1 doodle.canvas}
4 \width {400}
5 \height {400}
6
7 \class {hpane}
8 \name {doodle.ctr1}
9 \parent {doodle}
10 \children {doodle.ctr1.ops doodle.ctr1.brush doodle.ctr1.color}
11 \maxHeight {20}
12 \BGCOLOR {grey45}
13
14 \class {menu}
15 \name {doodle.ctr1.ops}
16 \parent {doodle.ctr1}
17 \menuConfig {
18   {backup}
19   {redraw}
20   {record on}
21   {record off}
22   {dump}
23   {clear stroke list}
24   {clearWindow}
25   {save}
26   {effect A, x10}
27   {effect A, x100}
28   {effect A, x200}
29   {effect A, x1000}
30   {effect B}
31   {effect C}
32   {effect D}
33   {effect E wave}
34 }
35 \label {Operation}
36 \maxWidth {100}
37 \FGColor {white}
38 \BGCOLOR {grey45}
39 \BDCOLOR {white}
40
41 \class {menu}
42 \name {doodle.ctr1.brush}
43 \parent {doodle.ctr1}
44 \menuConfig {
45   {line}
46   {point}
47   {larger point}
48   {even larger point}
49   {very large point}
50   {monster point}
51   {textual 1}
52   {textual 3}
53 }
54 \label {Brushes}
55 \maxWidth {100}
56 \FGColor {white}
57 \BGCOLOR {grey45}
58 \BDCOLOR {white}
59
60 \class {menu}
61 \name {doodle.ctr1.color}
62 \parent {doodle.ctr1}
63 \menuConfig {
64   {black}
65   {white}
66   {red}
67   {blue}
68   {green}
69 }
70 \label {Colors}
71 \maxWidth {100}
72 \FGColor {white}
73 \BGCOLOR {grey45}
74 \BDCOLOR {white}
75 \class {field}

```

```

76 \name {doodle.canvas}
77 \parent {doodle}
78 \script {
79   switch (arg[0]) {
80     case "mouseMove":
81       if (penDownP) {
82         x0 = x1;
83         y0 = y1;
84         x1 = arg[1];
85         y1 = arg[2];
86         if (brush == 0) {
87           drawLine(x0, y0, x1, y1);
88           if (record == 1) {
89             stroke[count] = concat("drawLine(",
90               x0, ",", y0, ",",
91               x1, ",", y1, ",");
92             count = count + 1;
93           }
94           } else if (brush == 1) {
95             drawLine(x1, y1, x1, y1);
96             if (record == 1) {
97               stroke[count] = concat("drawLine(",
98                 x1, ",", y1, ",",
99                 x1, ",", y1, ",");
100               count = count + 1;
101             }
102             } else if (brush == 2) {
103               drawFillRect(x1, y1, x1 + 2, y1 + 2);
104               if (record == 1) {
105                 stroke[count] = concat("drawFillRect(",
106                   x1, ",", y1, ",",
107                   x1 + 2, ",", y1 + 2, ",");
108                 count = count + 1;
109               }
110             } else if (brush == 3) {
111               drawFillRect(x1, y1, x1 + 5, y1 + 5);
112               if (record == 1) {
113                 stroke[count] = concat("drawFillRect(",
114                   x1, ",", y1, ",",
115                   x1 + 5, ",",
116                   y1 + 5, ",");
117                 count = count + 1;
118               }
119             } else if (brush == 4) {
120               drawFillRect(x1, y1, x1 + 10, y1 + 10);
121               if (record == 1) {
122                 stroke[count] = concat("drawFillRect(",
123                   x1, ",", y1, ",",
124                   x1 + 10, ",",
125                   y1 + 10, ",");
126                 count = count + 1;
127               }
128             } else if (brush == 5) {
129               drawFillRect(x1, y1, x1 + 100, y1 + 100);
130               if (record == 1) {
131                 stroke[count] = concat("drawFillRect(",
132                   x1, ",", y1, ",",
133                   x1 + 100, ",",
134                   y1 + 100, ",");
135                 count = count + 1;
136               }
137             } else if (brush == 6) {
138               c = nthChar(brushText, idx);
139               drawText(x1, y1, 1, c);
140               idx = idx + 1;
141               if (record == 1) {
142                 stroke[count] = concat("drawText(",
143                   x1, ",", y1, ",",
144                   "1, ", c, ",");
145                 count = count + 1;
146               }
147             } else if (brush == 7) {
148               c = nthChar(brushText, idx);
149               drawText(x1, y1, 3, c);
150             }

```



```

151     idx = idx + 1;
152     if (record == 1) {
153         stroke[count] = concat("drawText(",
154             x1, ", ", y1, ", ",
155             "3, ", c, ", ");
156         count = count + 1;
157     }
158     return;
159 }
160
161 case "buttonPress":
162     penDownP = 1;
163     x0 = mouseX();
164     y0 = mouseY();
165     x1 = x0;
166     y1 = y0;
167
168 break;
169 case "buttonRelease":
170     penDownP = 0;
171
172 break;
173 case "setColor":
174     set("FGColor", arg[1]);
175
176 break;
177 case "setBrush":
178     brush = arg[1];
179     if (brush == 6) {
180         brushText = loadFile("~/aliceInWonderland.txt");
181         brushText = loadFile("~/usr/work/viola/apps/doodle.v");
182     } else if (brush == 7) {
183         brushText = loadFile("~/usr/dict/words");
184     }
185     idx = 0;
186     brushText = loadFile("~/usr/dict/words");
187 }
188
189 break;
190 case "clear":
191     count = 0;
192     clearWindow();
193     return;
194
195 break;
196 case "clearWindow":
197     clearWindow();
198     return;
199
200 break;
201 case "save":
202     tt = "";
203     for (i = 0; i < count; i = i + 1) {
204         tt = concat(tt, stroke[i], "\n");
205     }
206     print("Saving to doodle.out\n");
207     saveFile("~/usr/tmp/doodle.out", tt);
208     return;
209
210 break;
211 case "dump":
212     for (i = 0; i < count; i = i + 1) {
213         print(i, "\t", stroke[i], "\n");
214     }
215     return;
216
217 break;
218 case "backup":
219     count = count - 1;
220     return;
221
222 break;
223 case "redraw":
224     tt = "";
225     for (i = 0; i < count; i = i + 1) {
226         tt = concat(tt, stroke[i]);
227     }
228     clearWindow();
229     interpret(tt);
230     return;
231
232 break;
233 case "record on":

```

```

234     record = 1;
235     return;
236
237 break;
238 case "record off":
239     record = 0;
240     return;
241
242 break;
243 case "effect A":
244     if (count <= 0) return;
245     n = arg[1];
246     for (i = 0; i < n; i = i + 1) {
247         idx = random(1) % count;
248         print("idx=", idx, "\n");
249         print("> ", idx, "\t", stroke[idx], "\n");
250         hue = (random(0) % 50) * 2;
251         set("FGColor", concat(hue, " ", hue, " ", hue));
252     }
253
254 break;
255 case "effect B":
256     if (count <= 0) return;
257     for (i = 0; i < count; i = i + 1) {
258         set("FGColor", "blue");
259         interpret(stroke[i]);
260     }
261     return;
262
263 break;
264 case "effect C":
265     if (count <= 0) return;
266     for (i = 0; i < count; i = i + 1) {
267         set("FGColor", "blue");
268         interpret(stroke[i]);
269     }
270     return;
271
272 break;
273 case "effect D":
274     if (count <= 0) return;
275     for (i = 0; i < count; i = i + 1) {
276         set("FGColor", "grey80");
277         interpret(stroke[i]);
278     }
279     return;
280
281 break;
282 case "effect E":
283     if (count <= 0) return;
284     for (i = 0; i < count; i = i + 1) {
285         set("FGColor", "grey70");
286         interpret(stroke[i]);
287     }
288     return;
289
290 break;
291 case "effect F":
292     if (count <= 0) return;
293     for (i = 0; i < count; i = i + 1) {
294         set("FGColor", "grey60");
295         interpret(stroke[i]);
296     }
297     return;
298
299 break;
300 case "effect G":
301     if (count <= 0) return;
302     for (i = 0; i < count; i = i + 1) {
303         set("FGColor", "grey50");
304         interpret(stroke[i]);
305     }
306     return;

```

```
301 case "effect E":
302   if (count <= 0) return;
303   for (i = 2; i < count; ) {
304     set("FGColor", "grey80");
305     interpret(stroke[i + 1]);
306
307     set("FGColor", "grey50");
308     interpret(stroke[i]);
309
310     set("FGColor", "grey80");
311     interpret(stroke[i - 1]);
312
313     set("FGColor", "white");
314     interpret(stroke[i - 2]);
315     i = i + 1;
316
317   }
318   return;
319 break;
320 case "init":
321   usual();
322   record = 1;
323   penDownP = 0;
324   return;
325 break;
326 }
327 usual();
328 }
```

```
1 \class {field}
2 \name {wave}
3 \parent {}
4 \children {wave.sb}
5 \script {}
6 switch (arg[0]) {
7   case "graph":
8     f = float(arg[1]);
9     xx = width();
10    r = height() / 2.0;
11    theta = 0;
12    for (x = 20; x < xx; x = x + 2) {
13      theta = theta + 0.1;
14      y = sin(theta * f) * r + r;
15      drawline(x, y, x, y + 1);
16    }
17    return;
18  break;
19 }
20 usual();
21 }
22 width {300}
23 height {200}
24 \BColor {blue}
25 \
26 \class {slider}
27 \name {wave.sb}
28 \parent {wave}
29 \script {
30   switch (arg[0]) {
31     case "_setPositionV":
32       usual();
33       send(parent(), "graph", arg[1]);
34       return;
35     break;
36   }
37   usual();
38 }
39 width {20}
40 height {200}
41 \showSizeV {10}
42 \BColor {white}
43 \BColor {grey45}
44 \FGColor {grey75}
45 \
46
47
```

```

1 \class {hpane}
2 \name {colorway}
3 \children {colorway.pic colorway.ctrl}
4 \script {
5   switch (arg[0]) {
6     case "findTop":
7       return send(parent(), "findTop");
8     break;
9   }
10  usual();
11 }
12 \BDCOLOR {LemonChiffon1}
13 \BGCOLOR {LemonChiffon1}
14 \width {400}
15 \height {200}
16 \
17 \class {field}
18 \name {colorway.pic}
19 \parent {colorway}
20 \script {
21   switch (arg[0]) {
22     case "graph":
23       theta0 = theta0 + 0.8;
24       theta1 = theta0 + 0.5;
25
26       c0 = c0 + dir;
27       if (c0 <= 0) {
28         dir = (random(0) % 10) + 10;
29       } else if (c0 > 99) {
30         dir = (random(0) % -10) - 10;
31       }
32       for (x = x0; x < x1; x = x + 2) {
33         theta0 = theta0 + 0.1;
34         y = sin(theta0 * f) * r + r;
35         theta1 = theta0 + 0.1;
36
37         set("FGColor", concat(c0, ' ', c0, ' ', c0));
38         drawLine(x, y, x, sin(theta1 * f) * r + r);
39
40       }
41       if (view == 1) after(100, self(), "graph");
42       return;
43     break;
44     case "clear":
45       clearWindow();
46       after(10000, self(), "clear");
47       return;
48     break;
49     case "randomizeBGColor":
50       if (view) {
51         set("BGColor", concatenate(random(0) % 100, " ",
52         random(0) % 100, " ",
53         random(0) % 100));
54       }
55       clearWindow();
56       after(10000, self(), "randomizeBGColor");
57
58     }
59   return;
60 }
61 case "VIEW_ON":
62   view = 1;
63   after(500, self(), "graph");
64   after(1, self(), "randomizeBGColor");
65 }
66 break;
67 case "expose":
68   send(send(parent(), "findTop"), "vw_relay", self(), "VIEW_ON");
69   send(send(parent(), "findTop"), "vw_relay", self(), "VIEW_OFF");
70   view = 1;
71   after(500, self(), "graph");
72   f = 10.0;
73   x0 = 0;
74   xx = width();
75   x1 = xx;
76   r = height() / 2.0;
77   after(1, self(), "randomizeBGColor");
78   after(500, self(), "graph");

```

```

76   after(10000, self(), "clear");
77 break;
78 case "VIEW_OFF":
79   view = 0;
80   return;
81 }
82 break;
83 case "buttonPress":
84   send(self(), "VIEW_ON");
85 break;
86 case "keyPress":
87   send(self(), "VIEW_OFF");
88 break;
89 }
90 usual();
91 }
92 \class {txtButton}
93 \parent {colorway}
94 \name {colorway.ctrl}
95 \label {New Color}
96 \script {
97   switch (arg[0]) {
98     case "buttonRelease":
99       send("colorway.pic", "randomizeBGColor");
100    break;
101  }
102  usual();
103 }
104 \maxWidth {100}
105 \maxHeight {25}
106 \gapH {5}
107 \BGCOLOR {grey45}
108 \BDCOLOR {white}
109 \

```

```

1 \class {vpane}
2 \name {vwq}
3 \children {vwq.about vwq.email vwq.name vwq.org vwq.listP vwq.blank vwq.send}
4 \width {400}
5 \height {400}
6 \script {
7   switch (arg[0]) {
8     case "queryAddress":
9       return "a viola app...";
10    break;
11    case "queryTitle":
12      return "a viola app title...";
13    break;
14    case "queryIsindex":
15      return 0;
16    break;
17    case "expose":
18      bell(2);
19      send(self(), "config", x(), y(), width(), height());
20    break;
21    default:
22      usual();
23  }
24 }
25 \class {txtDisp}
26 \name {vwq.about}
27 \parent {vwq}
28 \content {\f(3)ViolaWWW User Survey\f(1)}
29
30 Please take a few moments to fill out this
31 form for a users survey. Thanks.
32 }
33 \BGColor {LemonChiffon3}
34 \FGColor {black}
35 \
36 \class {hpane}
37 \name {vwq.email}
38 \parent {vwq}
39 \children {vwq.email.title vwq.email.entry}
40 \maxHeight {20}
41 \
42 \class {txtLabel}
43 \name {vwq.email.title}
44 \parent {vwq.email}
45 \label {Your E-Mail address:}
46 \paneConfig {eastToWest}
47 \maxWidth {150}
48 \BGColor {grey45}
49 \
50 \class {txtEdit}
51 \name {vwq.email.entry}
52 \parent {vwq.email}
53 \script {
54   switch (arg[0]) {
55     case "keyPress":
56       c = key();
57       switch (c) {
58         case '\t':
59           case '\r':
60             if (shiftKeyP()) vwq.send("focus");
61             return;
62             else vwq.name.entry("focus");
63           return;
64         default:
65           insert(c);
66           return;
67         break;
68       }
69     case "contentP":
70       return get("content");
71     break;
72     default:
73       usual();
74   }
75 }

```

```

76 \BGColor {LemonChiffon1}
77 \FGColor {black}
78 \CRColor {red}
79 \
80 \class {hpane}
81 \name {vwq.name}
82 \parent {vwq}
83 \children {vwq.name.title vwq.name.entry}
84 \maxHeight {20}
85 \
86 \class {txtEdit}
87 \name {vwq.name.entry}
88 \parent {vwq.name}
89 \script {
90   switch (arg[0]) {
91     case "keyPress":
92       c = key();
93       switch (c) {
94         case '\t':
95           case '\r':
96             if (shiftKeyP()) vwq.email.entry("focus");
97             else vwq.org.entry("focus");
98             return;
99         break;
100        default:
101          insert(c);
102          return;
103        break;
104      }
105    case "contentP":
106      return get("content");
107    break;
108    case "init":
109      set("content", pipe("whoami"));
110    /*
111    */
112    tmpWhoAmI = makeTempFile();
113    system(concat("whoami > ", tmpWhoAmI));
114    set("content", trimEdge(loadFile(tmpWhoAmI)));
115    system(concat("rm -f ", tmpWhoAmI));
116    /*
117    */
118    set("content", pipe("whoami"));
119    finfo = nthLine(pipe("finger 'whoami'", 1, 1));
120    cidx = findPattern(finfo, "In real life: ");
121    if (cidx != -1) {
122      set("content", nthChars(finfo, cidx, 999));
123    } else {
124      set("content", "");
125    }
126    /*
127    */
128    break;
129  }
130  usual();
131 }
132 \BGColor {LemonChiffon1}
133 \FGColor {black}
134 \CRColor {red}
135 \
136 \class {txtLabel}
137 \name {vwq.name.title}
138 \parent {vwq.name}
139 \label {Your name:}
140 \paneConfig {eastToWest}
141 \maxWidth {150}
142 \BGColor {grey45}
143 \
144 \class {hpane}
145 \name {vwq.org}
146 \parent {vwq}
147 \children {vwq.org.title vwq.org.entry}
148 \maxHeight {100}
149 \BGColor {grey45}
150 \
151 \class {txtLabel}
152 \name {vwq.org.title}

```

```

151 \parent {vwq.org}
152 \label {Your organization;}
153 \paneconfig {eastToWest}
154 \maxwidth {150}
155 \maxheight {20}
156 \BGcolor {grey45}
157 \
158 \class {txtEdit}
159 \name {vwq.org.entry}
160 \parent {vwq.org}
161 \script {
162     switch (arg[0]) {
163     case "keyPress":
164         c = key();
165         if (c == '\t') {
166             if (shiftKeyP()) vwq.name.entry("focus");
167             else vwq.listP.check("focus");
168             return;
169         } else {
170             insert(c);
171             return;
172         }
173     }
174     break;
175     case "contentP":
176         return get("content");
177     }
178     usual();
179 }
180 \BGcolor {LemonChiffon1}
181 \FGcolor {black}
182 \CRcolor {red}
183 \
184 \class {hpane}
185 \name {vwq.listP}
186 \parent {vwq}
187 \children {vwq.listP.check vwq.listP.title}
188 \maxheight {20}
189 \
190 \class {toggle}
191 \name {vwq.listP.check}
192 \parent {vwq.listP}
193 \script {
194     switch (arg[0]) {
195     case "keyPress":
196         c = key();
197         if (c == '\t') {
198             if (shiftKeyP()) vwq.org.entry("focus");
199             else vwq.blank("focus");
200             return;
201         } else {
202             send(self(), "buttonRelease");
203             return;
204         }
205     }
206     break;
207     case "contentP":
208         if (get("toggleState") == 1)
209             return "viola-announce@xcf.berkeley.edu";
210         else return "";
211     }
212     usual();
213 }
214 \style {check}
215 \maxwidth {18}
216 \BGcolor {grey45}
217 \
218 \class {txtLabel}
219 \name {vwq.listP.title}
220 \parent {vwq.listP}
221 \label {check to be added to viola-announce@xcf.berkeley.edu}
222 \script {
223     switch (arg[0]) {
224     case "buttonPress":
225     case "buttonRelease":

```

```

226     return send("vwq.listP.check", arg[0]);
227     break;
228     case "keyPress":
229         c = key();
230         if (c == '\t') {
231             if (shiftKeyP()) vwq.org.entry("focus");
232             else vwq.blank("focus");
233             return;
234         } else {
235             vwq.listP.check("buttonRelease");
236             return;
237         }
238     }
239     usual();
240 }
241 \font {normal}
242 \BGcolor {grey45}
243 \
244 \class {txtEdit}
245 \name {vwq.blank}
246 \parent {vwq}
247 \content {\Space for any comments you might have.}
248 \script {
249     switch (arg[0]) {
250     case "keyPress":
251         c = key();
252         switch (c) {
253         case '\t':
254             if (shiftKeyP()) vwq.listP.check("focus");
255             else vwq.send("focus");
256             return;
257         }
258         break;
259         default:
260             insert(c);
261             return;
262         }
263     }
264     break;
265     case "focus":
266         mousePos = mouse();
267         winPos = windowPosition();
268         mx = mousePos[0];
269         my = mousePos[1];
270         dx = (winPos[0] + width() / 2) - mx;
271         dy = (winPos[1] + height() / 2) - my;
272         for (i = 0; i < 10; i = i + 1) {
273             mx = mx + dx;
274             my = my + dy;
275             setMouse(mx, my);
276         }
277         return;
278     }
279     break;
280     case "contentP":
281         return get("content");
282     }
283     }
284     usual();
285 }
286 \BGcolor {LemonChiffon1}
287 \FGcolor {black}
288 \CRcolor {red}
289 \font {normal}
290 \
291 \class {txtButton}
292 \name {vwq.send}
293 \parent {vwq}
294 \label {Press to e-mail this off}
295 \script {
296     switch (arg[0]) {
297     case "buttonRelease":
298         tmpUserInfo = makeTempFile();
299         system(concat("whoami > ",
300

```

```

301 tmpWhoAmI = makeTempFile();
302 system(concat("who am i > ", tmpWhoAmI));
303
304 /**bugged
305 "User Info:: ", pipe("grep 'whoami' /etc/passwd"), "\n"
306 "WhoAmI:: ", pipe("who am i"), "\n",
307 */
308
309 tt = concat(
310     "**** Survey extract *****\n",
311     "User email : ", vwq.email.entry("contentP"), "\n",
312     "User name : ", vwq.name.entry("contentP"), "\n",
313     "Add to : ", vwq.listP.check("contentP"), "\n",
314     "Organiz. : ", vwq.org.entry("contentP"), "\n",
315     "Who_Am_I : ", trimEdge(loadFile(tmpWhoAmI)), "\n",
316     "User Info : ", trimEdge(loadFile(tmpUserInfo)), "\n",
317     "Comments : ", vwq.blank("contentP"), "\n",
318     "*****\n");
319
320 print("\n\nIf this survey were for real, the following text would
be mailed:\n\n", tt, "\n\n");
321
322 tmp = makeTempFile();
323 saveFile(tmp, tt);
324 system(concat(MAILER, " ", "-s survey ",
RECEIVER, " < ", tmp));
325
326
327 system(concat("rm -f ", tmp,
" ", tmpUserInfo,
" ", tmpWhoAmI
));
328
329
330
331 break;
332 case "keyPress":
333     c = key();
334     if (c == '\t') {
335         if (shiftKeyP()) vwq.blank("focus");
336         else vwq.email.entry("focus");
337         return;
338     } else if (c == '\r') {
339         send(self(), "buttonRelease");
340         return;
341     }
342 break;
343 case "init":
344     RECEIVER = "pei@pebble";
345     MAILER = "Mail";
346 break;
347 }
348 usual();
349
350 } \BCColor {grey45}
351 \BDColor {white}
352 \maxHeight {25}
353 \gapH {2}
354 \gapV {3}
355

```



**ViolaWWW Hypermedia Browser**  
file://client/home/wei/violaTOGO/www/docs/about\_localVersion.hmml



# ViolaWorldWideWeb Hypermedia Browser

1993.3.7

ViolaWWW implements an XWindows browser to the World Wide Web system.

This browser has the basic features of of hypertext navigation mechanisms: clickable hot-words, history list. It has keyword search entry field, and can do page cloning (multiple windows). There exists a simple printer hook, and a document source viewer.

This version of ViolaWWW support two separate DTDs -- HTML (which includes extensions such as inlined images) and HMML. A simple stylesheet mechanism is supported. The book-mark lists have been deprecated.

ViolaWWW is built using the Viola hypermedia toolkit, and now also comes with a Motif front end.

**See the demo documents.**

Connect to the Viola Central for more info regarding this software.

---

## Acknowledgements

The Experimental Computing Facility (XCF) at U.C. Berkeley supported the initial work on the Viola project. Current work on Viola and its applications is supported by O'Reilly & Associates.

Most importantly, my sincere thanks go to all the people who've contributed to Viola in various ways -- with bug reports, suggestions, code patches, moral encouragements, etc. You know who you are. Thanks.

---

## Contacts

Please address to me the ViolaWWW browser specific comments and questions. There is also a www-talk@info.cern.ch mailing list, and a comp.infosystems.www news group.

Pei Y. Wei

Search:

Back Up

Previous

Next

Show Source

Clone Page





# Test misc

http://server/home/wei/violaTOGO/docs/tests/testAll.html

## Header 1

## Header 2

### Header 3

See the [motd](#) for useless messages. Here is an ordered list:

- 1 socks
- 2 scarf
- 3 pillows

And an unordered list.

- rubber ball.
- I recollect that, when a stripling, my first exploit in squirrel-shooting was in a grove of tall walnut-trees that shades one side of the valley. I had wandered into it at noontime, when all nature is peculiarly quiet, and was startled by the roar of my own gun, as it broke the Sabbath stillness around and was prolonged and reverberated by the angry echoes.
- If ever I should wish for a retreat whither I might steal from the world and its distractions, and dream quietly away the remnant of a troubled life, I know of none more promising than this little valley.

Definition lists:

First term

Definition paragraph starts here.

Second term

Second definition paragraph starts here.

Here is a dir (x2):

socks 0

scarf 1

Search:

[Back Up](#)[Previous](#)[Next](#)[Show Source](#)[Clone Page](#)



## test input

<http://server/home/wei/violaTOGO/docs/tests/testInputAll.html>

# FORM samples

This document contains a variety of forms as supported by viola, in HMML.

Please enter your name:

age:

Sex:

 Male Female

## Pizza sizes

 small medium large

## Toppings

add 25 cents for each topping

 Anchovies Salami Sausage Green peppers Mushrooms Artichokes

Red hot pepper density: (number of shakes -- 9 is ideal)

Cheese density: (50% is recommended)

Tomato sauce depth: (in millimeters. 3 is nice and moist)

Search:



# test input

http://server/home/wei/violaTOGO/docs/tests/testInputAll.html

add 25 cents for each topping

- Anchovies
- Salami
- Sausage
- Green peppers
- Mushrooms
- Artichokes

Red hot pepper density: (number of shakes -- 9 is ideal)

99

Cheese density: (50% is recommended)

50

Tomato sauce depth: (in millimeters, 3 is nice and moist)

3

## Option menu

- ▼ San Francisco Which city are you going to? aftereffect afterglow afterimage afterlife aftermath afternoon afterthought afterward afterword again against Agamemnon agate Agatha agave age

Submit Form

Radio buttons.

## Signature

Clear pad

*Scott*

Submit Form

Search:

Back Up

Previous

Next

Show Source

Clone Page



# RandoMyster's Waves

<http://server/home/wei/violaTOGO/docs/gallery/randommyster.html>

**Title**

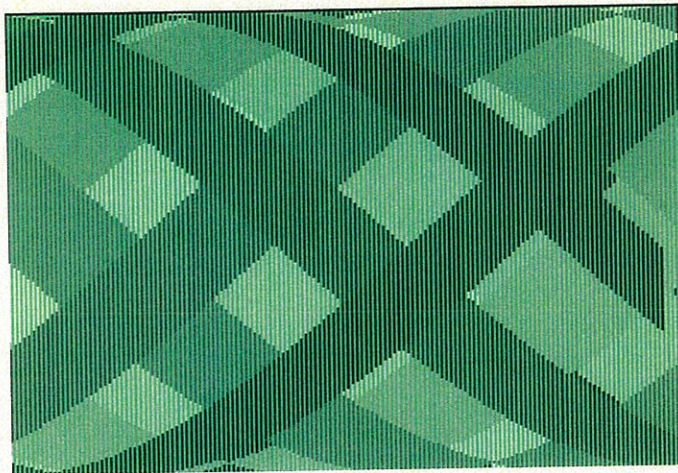
Brush Motion

**Artist**

Rando "random art does not apologize" Myster

**Date**

1993, March 7, before breakfast



New Color

Warning: playing with color too much will take all the color resources... (Yes, this is a bug).

Search:

[Back Up](#)

[Previous](#)

[Next](#)

[Show Source](#)

[Clone Page](#)



## Embedded Viola objects

http://server/home/wei/violaTOGO/docs/viola/violaApps.html

# Embedding Viola Objects in HMML Document

The <VDBJF> tag can be used to insert viola applications into a document. Since a viola objects are programmed, this is also a way to link programming into the document.

The <VINSERT> tag has similar functionality, but executes the script and inserts the result of the execution.

Of course, too much violaism reduces the portability of your document on the World Wide Web. But anyway, this is neat... This tag should be deprecated and use <EMBED> instead, and treat viola objects as foreign data types, like postscript.

Ideas for dealing with security include: disabling these objects' privilege to execute system sensitive commands such as `system()`; Limit the scope of the messaging to be only among the objects related to the document... This is an open question, so I (pei@ora.com) would be interested in ideas.

Here are a couple of examples of embedded viola applications (objects).

## User Survey

This form isn't done with HTML+ input form. It's a a viola application that's a form. This particular example is obsolete since the advent of HTML+ forms, but potentially this is useful for building arbitrary new form devices that aren't in the HTML+ standard.

### ViolaWWW User Survey

Please take a few moments to fill out this form for a users survey. Thanks.

Your E-Mail address:	<input type="text"/>
Your name:	wei
Your organization:	<input type="text"/>

Search:

[Back Up](#)[Previous](#)[Next](#)[Show Source](#)[Clone Page](#)



## Embedded Viola objects

<http://server/home/wei/violaTOGO/docs/viola/violaApps.html>

Please take a few moments to fill out this form for a users survey. Thanks.

Your E-Mail address:

Your name: wei

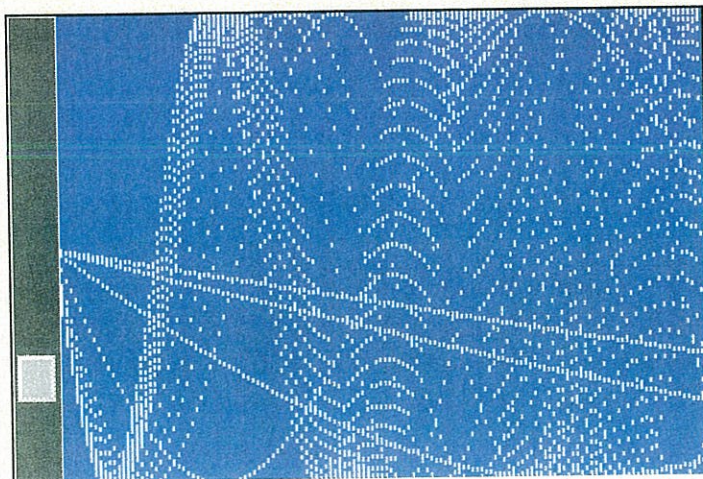
Your organization:

Check to be added to [viola-announce@xcf.berkeley.edu](mailto:viola-announce@xcf.berkeley.edu)  
Space for any comments you might have.

Press to e-mail this off

## Wave motion

More useful would be some kind of interactive equation plotter.



Search:

:

Back Up

Previous

Next

Show Source

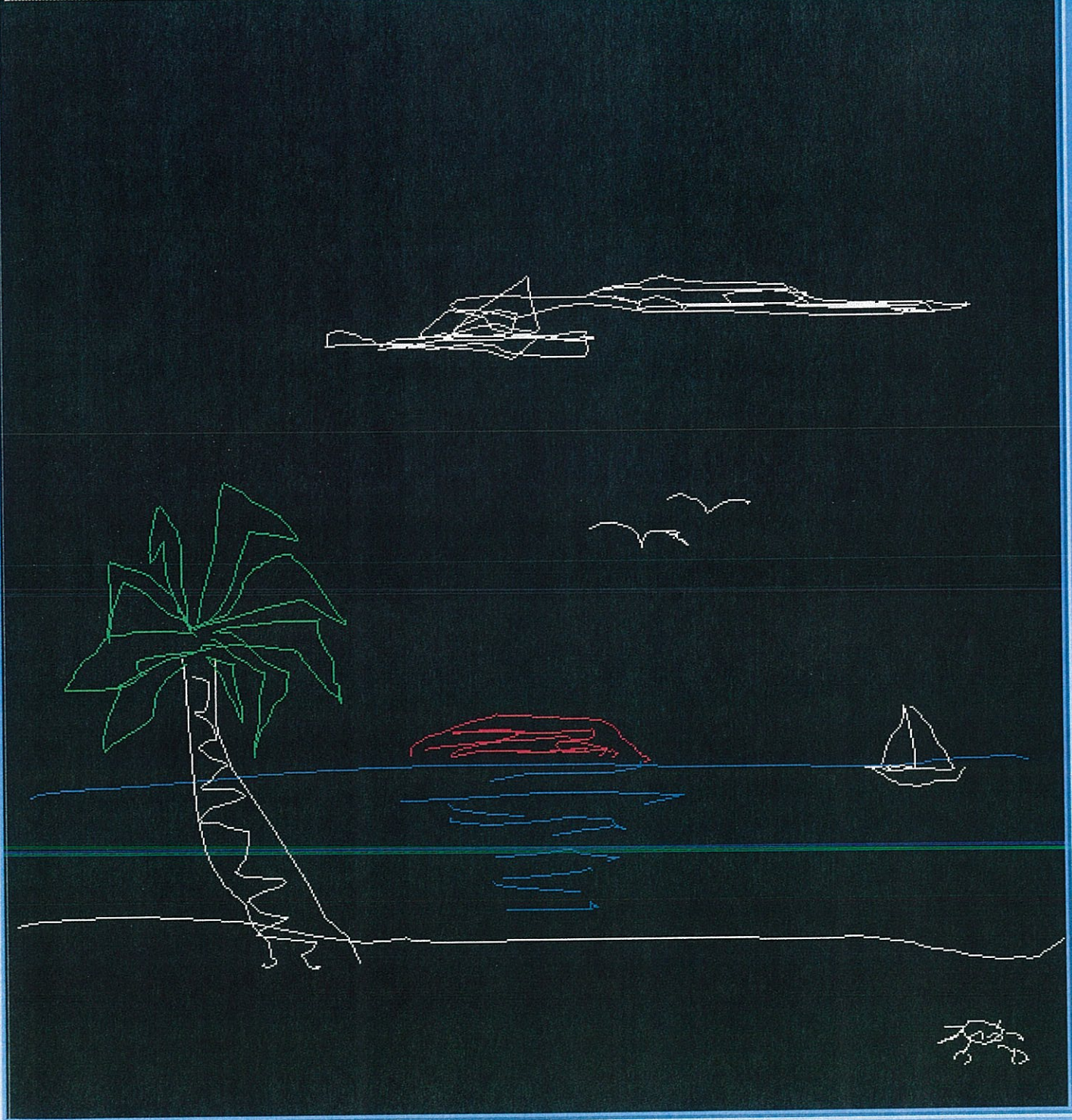
Clone Page



# doodle

http://server/home/wei/violaTOGO/demo/objs/doodle.v

▼ Operation    ▼ Brushes    ▼ Colors



Search:

Back Up

Previous

Next

Show Source

Clone Page