

```
1 /*****
2 //
3 // UNIFORM BUILDER README:
4 //
5 // The script in this frame performs only some basic inits & processing.
6 // In general, the bulk of the code is contained in frame("script") #25.
7 // Some additional code is scattered in some of the buttons and a few of
8 // the controls.
9 //
10 // FILES REQUIRED @ RUNTIME:
11 // login.asp, load.asp, loadUID.asp, delete.asp, save.asp, urls.txt
12 //
13 // FILES REQUIRED @ GENTIME:
14 // custom-football.as
15 //
16 /*****/
17
18 // load urls for button links: membersLink, dealersLink, builderLink
19 loadVariables("urls.txt", this);
20
21 user = "";
22 password = "";
23
24 timeout = 30000;
25 uniIndex = -1;
26 start = getTimer();
27 totalBytes = _root.getBytesTotal();
28 notSavedUniforms = new Array();
29 savedUniforms = new Array();
30 uniformName = "";
31 xmlReady = false;
32 showLogin = true;
33
34 CODEA = "A".charCodeAt(0);
35 CODEZ = "Z".charCodeAt(0);
36 CODE0 = "0".charCodeAt(0);
37 CODE9 = "9".charCodeAt(0);
38
39 function getUniform(node)
40 {
41     var theAttributes = node.attributes;
42     for (props in theAttributes)
43     {
44         var theName = "theAttributes."+props;
45         var theValue = eval(theName);
46         if ((theValue.charCodeAt(0) >= CODE0) && (theValue.charCodeAt(0) <= CODE9) && (props!="UID"))
47             eval(theName) = parseInt(theValue);
48         else if (theValue == "true")
49             eval(theName) = true;
50         else if (theValue == "false")
51             eval(theName) = false;
52     }
53
54     return theAttributes;
55 }
56
57 function getSavedList()
58 {
59     if ((!this.loaded) || (this.status != 0))
60     {
61         // error in xml, move on w/o saved list
62         xmlReady = true;
63         return;
64     }
65
66     var top = this.lastChild;
67     while (top.nodeName != "savedList")
68     {
69         top = top.previousSibling;
70     }
71
72     top = top.firstChild;
73     i=0;
74     while (top.nodeName == "uniform")
75     {
76         var theUniform = getUniform(top);
77
78         if (theUniform.UniNum == 0)
79         {
80             notSavedUniforms.push(theUniform);
81         }
82         else
83         {
84             theUniform.uniIndex = i;
```

```
85         savedList.addItem(theUniform);
86         savedUniforms.push(theUniform);
87         i++;
88     }
89
90     top = top.nextSibling;
91 }
92
93 if (top.nodeName == "messages")
94 {
95     var theAttributes = top.attributes;
96     for (props in theAttributes)
97     {
98         eval(props) = eval("theAttributes."+props);
99     }
100 }
101
102 xmlReady = true;
103 }
104
105 // extend string methods
106 String.prototype.contains = function(aString) { return (this.indexOf(aString) >=0); }
107 String.prototype.startsWith = function(aString) { return (this.indexOf(aString) == 0); }
108 String.prototype.endsWith = function(aString) { return (this.contains(aString) && (this.indexOf(aString) == (this.le
    ngth-aString.length))); }
109 function isDefined(theValue) { return typeof(theValue) != "undefined"; }
110
111 function initBuilder()
112 {
113     // whether within permitted online ordering dates
114     if (isDefined(orderAllowed))
115     {
116         orderAllowed = true;
117     }
118
119     if (isDefined(orderDeadline))
120     {
121         orderDeadline = "01/01/03";
122     }
123     orderDeadline = "Your order cannot be guaranteed\nunless it is placed by "+orderDeadline;
124
125     // if loading from another builder or UID supplied, do not redisplay login screen
126     if (isDefined(UID))
127     {
128         showLogin = false;
129     }
130     else
131     {
132         UID = "";
133     }
134
135     if (isDefined(CusNum))
136     {
137         CusNum = parseInt(CusNum);
138         showLogin = false;
139     }
140     else
141     {
142         CusNum = -1;
143     }
144
145     if (!showLogin)
146     {
147         theParams = new XML();
148         theElement = theParams.createElement("params");
149         theElement.attributes.UID = UID;
150         theElement.attributes.CusNum = CusNum;
151         theParams.appendChild(theElement);
152
153         thexml = new XML();
154         thexml.onLoad = getSavedList;
155         theParams.sendAndLoad("load.asp", thexml);
156
157         // wait until login screen loaded to bypass
158         stop();
159     }
160     else
161     {
162         // move onto login screen
163         play();
164     }
165 }
166
167 initBuilder();
```

```
1 BASE_OFFSET = 5;
2 BASE_OFFSET2 = Math.pow(BASE_OFFSET,2);
3
4 // jerseyCodes[_jerseyConstruct][_jerseyStyle][5*_jerseyStyleExtra + _jerseyFabric];
5 jerseyCodes = new Array();
6 jerseyCodes[ELITE] = [[],[],[],[],[],[],[],[],[],[],[]];
7 jerseyCodes[VARSITY] = [[],[],[],[],[],[],[],[],[],[],[]];
8
9 // jerseyPrices[_jerseyConstruct][_jerseyStyle][BASE_OFFSET*_jerseyStyleExtra + _jerseyFabric];
10 jerseyPrices = new Array();
11 jerseyPrices[ELITE] = [[],[],[],[],[],[],[],[],[],[],[]];
12 jerseyPrices[VARSITY] = [[],[],[],[],[],[],[],[],[],[],[]];
13
14 // pantCodes[_pantConstruct][_pantStyle][(_waistband?1:0)*100 + (_pantSide1Pattern==NONE)?1:0)*10 + _pantStyleExtra ]
15 ;
16 pantCodes = new Array();
17 pantCodes[ELITE] = [[],[]];
18 pantCodes[VARSITY] = [[],[]];
19
20 // pantPrices[_pantConstruct][_pantStyle][(_waistband?1:0)*100 + (_pantSide1Pattern==NONE)?1:0)*10 + _pantStyleExtra ]
21 ;
22 pantPrices = new Array();
23 pantPrices[ELITE] = [[],[]];
24 pantPrices[VARSITY] = [[],[]];
25
26 setJerseyCode(ELITE, TERPS, NONE, PROBRITE, "NJAU07301");
27 setJerseyCode(ELITE, TERPS, NONE, PORTHOLE, "NJAU07302");
28 setJerseyCode(ELITE, TERPS, CAPSLEEVE, PROBRITE, "NJAU07446");
29 setJerseyCode(ELITE, TERPS, CAPSLEEVE, PORTHOLE, "NJAU07447");
30 setJerseyCode(ELITE, HURRICANES, NONE, PROBRITE, "NJAU07295");
31 setJerseyCode(ELITE, HURRICANES, NONE, PORTHOLE, "NJAU07296");
32 setJerseyCode(ELITE, HURRICANES, CAPSLEEVE, PROBRITE, "NJAU07295");
33 setJerseyCode(ELITE, HURRICANES, CAPSLEEVE, PORTHOLE, "NJAU07440");
34 setJerseyCode(ELITE, USC, NONE, PROBRITE, "NJAU07299");
35 setJerseyCode(ELITE, USC, NONE, PORTHOLE, "NJAU07300");
36 setJerseyCode(ELITE, USC, CAPSLEEVE, PROBRITE, "NJAU07442");
37 setJerseyCode(ELITE, USC, CAPSLEEVE, PORTHOLE, "NJAU07443");
38 setJerseyCode(ELITE, DUCKS, NONE, PROBRITE, "NJAU07297");
39 setJerseyCode(ELITE, DUCKS, NONE, PORTHOLE, "NJAU07298");
40 setJerseyCode(ELITE, DUCKS, CAPSLEEVE, PROBRITE, "NJAU07444");
41 setJerseyCode(ELITE, DUCKS, CAPSLEEVE, PORTHOLE, "NJAU07445");
42 setJerseyCode(ELITE, BATWING, NONE, PROBRITE, "NJGM01426");
43 setJerseyCode(ELITE, BATWING, NONE, PORTHOLE, "NJGM03424");
44 setJerseyCode(ELITE, BATWING, CAPSLEEVE, PROBRITE, "NJGM04448");
45 setJerseyCode(ELITE, BATWING, CAPSLEEVE, PORTHOLE, "NJGM04809");
46 setJerseyCode(ELITE, BATWING, BELT_LEN, PORTHOLE, "NJGM05619");
47 setJerseyCode(ELITE, SHOULDER_INS, NONE, PROBRITE, "NJGM04548");
48 setJerseyCode(ELITE, SHOULDER_INS, NONE, PORTHOLE, "NJGM04810");
49 setJerseyCode(ELITE, SHOULDER_INS, BELT_LEN, PROBRITE, "NJGM07197");
50 setJerseyCode(ELITE, LINEMAN, NONE, PROBRITE, "NJGM01424");
51 setJerseyCode(ELITE, LINEMAN, NONE, PORTHOLE, "NJGM01428");
52 setJerseyCode(ELITE, LINEMAN, CAPSLEEVE, PROBRITE, "NJGM01416");
53 setJerseyCode(ELITE, LINEMAN, CAPSLEEVE, PORTHOLE, "NJGM03427");
54 setJerseyCode(ELITE, LINEMAN, BELT_LEN, PROBRITE, "NJGM020BL");
55 setJerseyCode(ELITE, SKILL, NONE, PROBRITE, "NJGM01430");
56 setJerseyCode(ELITE, SKILL, NONE, PORTHOLE, "NJGM03428");
57 setJerseyCode(ELITE, SKILL, BELT_LEN, PROBRITE, "NJGM01866");
58 setJerseyCode(ELITE, MESH, NONE, PROBRITE, "NJGM01434");
59 setJerseyCode(VARSITY, LINEMAN, NONE, PROBRITE, "NJGM03429");
60 setJerseyCode(VARSITY, LINEMAN, NONE, PORTHOLE, "NJGM03438");
61 setJerseyCode(VARSITY, LINEMAN, CAPSLEEVE, PROBRITE, "NJGM03433");
62 setJerseyCode(VARSITY, LINEMAN, CAPSLEEVE, PORTHOLE, "NJGM03437");
63 setJerseyCode(VARSITY, LINEMAN, BELT_LEN, PROBRITE, "NJGM07198");
64 setJerseyCode(VARSITY, SKILL, NONE, PROBRITE, "NJGM03431");
65 setJerseyCode(VARSITY, SKILL, NONE, PORTHOLE, "NJGM03435");
66 setJerseyCode(VARSITY, SKILL, BELT_LEN, PROBRITE, "NJGM03432");
67 setJerseyCode(VARSITY, SKILL, BELT_LEN, PORTHOLE, "NJGM03436");
68 setJerseyCode(VARSITY, MESH, NONE, PROBRITE, "NJGM03430");
69
70 setJerseyPrice(ELITE, TERPS, NONE, PROBRITE, 54.50);
71 setJerseyPrice(ELITE, TERPS, NONE, PORTHOLE, 54.50);
72 setJerseyPrice(ELITE, TERPS, CAPSLEEVE, PROBRITE, 54.50);
73 setJerseyPrice(ELITE, TERPS, CAPSLEEVE, PORTHOLE, 54.50);
74 setJerseyPrice(ELITE, HURRICANES, NONE, PROBRITE, 70.00);
75 setJerseyPrice(ELITE, HURRICANES, NONE, PORTHOLE, 70.00);
76 setJerseyPrice(ELITE, HURRICANES, CAPSLEEVE, PROBRITE, 70.00);
77 setJerseyPrice(ELITE, HURRICANES, CAPSLEEVE, PORTHOLE, 70.00);
78 setJerseyPrice(ELITE, USC, NONE, PROBRITE, 55.50);
79 setJerseyPrice(ELITE, USC, NONE, PORTHOLE, 55.50);
80 setJerseyPrice(ELITE, USC, CAPSLEEVE, PROBRITE, 55.50);
81 setJerseyPrice(ELITE, USC, CAPSLEEVE, PORTHOLE, 55.50);
82 setJerseyPrice(ELITE, DUCKS, NONE, PROBRITE, 49.00);
83 setJerseyPrice(ELITE, DUCKS, NONE, PORTHOLE, 49.00);
84 setJerseyPrice(ELITE, DUCKS, CAPSLEEVE, PROBRITE, 49.00);
85 setJerseyPrice(ELITE, DUCKS, CAPSLEEVE, PORTHOLE, 49.00);
86
87 setJerseyPrice(VARSITY, LINEMAN, NONE, PROBRITE, 54.50);
88 setJerseyPrice(VARSITY, LINEMAN, NONE, PORTHOLE, 54.50);
89 setJerseyPrice(VARSITY, LINEMAN, CAPSLEEVE, PROBRITE, 54.50);
90 setJerseyPrice(VARSITY, LINEMAN, CAPSLEEVE, PORTHOLE, 54.50);
91 setJerseyPrice(VARSITY, LINEMAN, BELT_LEN, PROBRITE, 70.00);
92 setJerseyPrice(VARSITY, LINEMAN, BELT_LEN, PORTHOLE, 70.00);
93 setJerseyPrice(VARSITY, SKILL, NONE, PROBRITE, 55.50);
94 setJerseyPrice(VARSITY, SKILL, NONE, PORTHOLE, 55.50);
95 setJerseyPrice(VARSITY, SKILL, BELT_LEN, PROBRITE, 55.50);
96 setJerseyPrice(VARSITY, SKILL, BELT_LEN, PORTHOLE, 55.50);
97 setJerseyPrice(VARSITY, MESH, NONE, PROBRITE, 49.00);
98 setJerseyPrice(VARSITY, MESH, NONE, PORTHOLE, 49.00);
99 setJerseyPrice(VARSITY, MESH, CAPSLEEVE, PROBRITE, 49.00);
100 setJerseyPrice(VARSITY, MESH, CAPSLEEVE, PORTHOLE, 49.00);
```

```
83 setJerseyPrice(ELITE, DUCKS, CAPSLEEVE, PORTHOLE, 49.00);
84 setJerseyPrice(ELITE, BATWING, NONE, PROBRITE, 49.00);
85 setJerseyPrice(ELITE, BATWING, NONE, PORTHOLE, 49.00);
86 setJerseyPrice(ELITE, BATWING, CAPSLEEVE, PROBRITE, 47.00);
87 setJerseyPrice(ELITE, BATWING, CAPSLEEVE, PORTHOLE, 47.00);
88 setJerseyPrice(ELITE, BATWING, BELT_LEN, PORTHOLE, 41.00);
89 setJerseyPrice(ELITE, SHOULDER_INS, NONE, PROBRITE, 46.00);
90 setJerseyPrice(ELITE, SHOULDER_INS, NONE, PORTHOLE, 46.00);
91 setJerseyPrice(ELITE, SHOULDER_INS, BELT_LEN, PROBRITE, 38.00);
92 setJerseyPrice(ELITE, LINEMAN, NONE, PROBRITE, 41.50);
93 setJerseyPrice(ELITE, LINEMAN, NONE, PORTHOLE, 41.50);
94 setJerseyPrice(ELITE, LINEMAN, CAPSLEEVE, PROBRITE, 41.50);
95 setJerseyPrice(ELITE, LINEMAN, CAPSLEEVE, PORTHOLE, 41.50);
96 setJerseyPrice(ELITE, LINEMAN, BELT_LEN, PROBRITE, 40.00);
97 setJerseyPrice(ELITE, SKILL, NONE, PROBRITE, 34.50);
98 setJerseyPrice(ELITE, SKILL, NONE, PORTHOLE, 34.50);
99 setJerseyPrice(ELITE, SKILL, BELT_LEN, PROBRITE, 32.00);
100 setJerseyPrice(ELITE, MESH, NONE, PROBRITE, 33.00);
101 setJerseyPrice(VARSITY, LINEMAN, NONE, PROBRITE, 33.50);
102 setJerseyPrice(VARSITY, LINEMAN, NONE, PORTHOLE, 33.50);
103 setJerseyPrice(VARSITY, LINEMAN, CAPSLEEVE, PROBRITE, 34.50);
104 setJerseyPrice(VARSITY, LINEMAN, CAPSLEEVE, PORTHOLE, 34.50);
105 setJerseyPrice(VARSITY, LINEMAN, BELT_LEN, PROBRITE, 33.50);
106 setJerseyPrice(VARSITY, SKILL, NONE, PROBRITE, 27.00);
107 setJerseyPrice(VARSITY, SKILL, NONE, PORTHOLE, 27.00);
108 setJerseyPrice(VARSITY, SKILL, BELT_LEN, PROBRITE, 25.40);
109 setJerseyPrice(VARSITY, SKILL, BELT_LEN, PORTHOLE, 25.50);
110 setJerseyPrice(VARSITY, MESH, NONE, PROBRITE, 26.00);
111
112 setPantCode(ELITE, TAPERED, 0, 0, 3, "NPKN07304");
113 setPantCode(ELITE, TAPERED, 1, 0, 3, "NPKN07305");
114 setPantCode(ELITE, TUNNEL, 0, 0, 3, "NPKN07306");
115 setPantCode(ELITE, TUNNEL, 1, 0, 3, "NPKN07307");
116 setPantCode(ELITE, TUNNEL, 0, 1, 3, "NPKN07308");
117 setPantCode(ELITE, TUNNEL, 1, 1, 3, "NPKN07309");
118 setPantCode(ELITE, TUNNEL, 0, 0, 5, "NPKN07312");
119 setPantCode(ELITE, TUNNEL, 1, 0, 5, "NPKN07313");
120 setPantCode(ELITE, TUNNEL, 0, 1, 5, "NPKN07310");
121 setPantCode(ELITE, TUNNEL, 1, 1, 5, "NPKN07311");
122 setPantCode(VARSITY, TUNNEL, 0, 0, 3, "NPGM03439");
123 setPantCode(VARSITY, TUNNEL, 1, 0, 3, "NPGM03440");
124
125 setPantPrice(ELITE, TAPERED, 0, 0, 3, 48.00);
126 setPantPrice(ELITE, TAPERED, 1, 0, 3, 50.00);
127 setPantPrice(ELITE, TUNNEL, 0, 0, 3, 43.00);
128 setPantPrice(ELITE, TUNNEL, 1, 0, 3, 45.00);
129 setPantPrice(ELITE, TUNNEL, 0, 1, 3, 49.00);
130 setPantPrice(ELITE, TUNNEL, 1, 1, 3, 51.00);
131 setPantPrice(ELITE, TUNNEL, 0, 0, 5, 48.00);
132 setPantPrice(ELITE, TUNNEL, 1, 0, 5, 50.00);
133 setPantPrice(ELITE, TUNNEL, 0, 1, 5, 51.00);
134 setPantPrice(ELITE, TUNNEL, 1, 1, 5, 53.00);
135 setPantPrice(VARSITY, TUNNEL, 0, 0, 3, 37.50);
136 setPantPrice(VARSITY, TUNNEL, 1, 0, 3, 39.50);
137
138 // [font][solid..accent]
139 numPrintPrice = [[ 1.85, 3.50 ],[ 2.05, 3.80 ]];
140 namePrintPrice = [ 1.54, 3.15 ];
141
142 // [size][solid,outline,shadow]
143 numTwillPrice = new Array();
144 numTwillPrice[4] = [ 1.40, 2.50, 2.50 ];
145 numTwillPrice[10] = [ 2.40, 4.50, 4.50 ];
146 numTwillPrice[12] = [ 2.80, 5.35, 5.35 ];
147
148 nameTwillPrice = new Array();
149 nameTwillPrice[2] = [ 1.25, 2.30, 2.30 ];
150 nameTwillPrice[3] = [ 1.35, 2.50, 2.50 ];
151
152 // options
153 miteredNeckPrice = 1.90;
154 jerseySideColorPrice = 1.80;
155 jerseySidePatternPrice = 2.00;
156 homeplatePrice = 2.20;
157 jerseyBraidPrice = 5.00;
158 pantBraidPrice = 7.50;
159 beltPrice = 5.00;
160 cuffVarsityPrice = 2.50;
161 cuffElitPrice = 4.20;
162 sleevePatternPrice = 3.70;
163
```

```
219 //stripe pattern options
220 SLV_WIDTH=5.5;
221 MAX_PATTERN_COLORS = 3;
222
223 sleevePatternNames = [
224     "323E", "33E", "23BC", "422", "335E", "435E",
225     "NW", "35F", "A7711", "A1411", "GB1410", "3NTS",
226     "IOWA", "535E", "NWF", "NW73", "2NKNW", "3NKNW"
227 ];
228
229 sleevePatternWidths = new Array();
230 sleevePatternWidths.SLV_323E = [ 1/1, 1/1, 1/1 ];
231 sleevePatternWidths.SLV_33E = [ 1/1, 1/1, 1/1 ];
232 sleevePatternWidths.SLV_23BC = [ 1/1, 3/2, 1/1 ];
233 sleevePatternWidths.SLV_422 = [ 2/1, 2/1 ];
234 sleevePatternWidths.SLV_335E = [ 1/2, 1/2, 1/2, 1/2, 1/2 ];
235 sleevePatternWidths.SLV_435E = [ 3/4, 3/4, 1/2, 3/4, 3/4 ];
236 sleevePatternWidths.SLV_NW = [ 1/2, 1/2, 2/1, 1/2, 1/2 ];
237 sleevePatternWidths.SLV_35F = [ 3/4, 1/4, 2/1, 1/4, 3/4 ];
238 sleevePatternWidths.SLV_A7711 = [ 1/8, 3/4, 1/8, 3/4, 1/8, 3/4, 1/8, 3/4, 1/8, 3/4, 1/8 ];
239 sleevePatternWidths.SLV_A1411 = [ 1/4, 1/1, 1/4, 1/8, 1/4, 1/1, 1/4, 1/8, 1/4, 1/1, 1/4 ];
240 sleevePatternWidths.SLV_GB1410 = [ 1/1, 1/4, 1/2, 1/4, 1/1, 1/4, 1/2, 1/4, 1/1 ];
241 sleevePatternWidths.SLV_3NTS = [ 1/2, 1/1, 1/2, 1/1, 1/2 ];
242 sleevePatternWidths.SLV_IOWA = [ 1/2, 1/4, 1/2, 1/4, 2/1, 1/4, 1/2, 1/4, 1/2 ];
243 sleevePatternWidths.SLV_535E = [ 1/1, 1/1, 1/1, 1/1, 1/1 ];
244 sleevePatternWidths.SLV_NWF = [ 1/4, 1/2, 1/4, 1/2, 1/4, 2/1, 1/4, 1/2, 1/4, 1/2, 1/4 ];
245 sleevePatternWidths.SLV_NW73 = [ 1/2, 1/2, 1/2, 1/2, 2/1, 1/2, 1/2, 1/2, 1/2 ];
246 sleevePatternWidths.SLV_2NKNW = [ 1/2, 1/2, 1/1, 1/2, 1/2 ];
247 sleevePatternWidths.SLV_3NKNW = [ 1/2, 1/2, 1/1, 1/2, 1/2 ];
248
249 sleevePatternColors = new Array();
250 sleevePatternColors.SLV_323E = [ 1, 2, 1 ];
251 sleevePatternColors.SLV_33E = [ 1, 3, 2 ];
252 sleevePatternColors.SLV_23BC = [ 1, 2, 1 ];
253 sleevePatternColors.SLV_422 = [ 1, 2 ];
254 sleevePatternColors.SLV_335E = [ 1, 0, 2, 0, 1 ];
255 sleevePatternColors.SLV_435E = [ 1, 0, 2, 0, 1 ];
256 sleevePatternColors.SLV_NW = [ 1, 0, 2, 0, 1 ];
257 sleevePatternColors.SLV_35F = [ 1, 0, 2, 0, 1 ];
258 sleevePatternColors.SLV_A7711 = [ 1, 2, 1, 0, 1, 3, 1, 0, 1, 2, 1 ];
259 sleevePatternColors.SLV_A1411 = [ 1, 2, 1, 0, 1, 2, 1, 0, 1, 2, 1 ];
260 sleevePatternColors.SLV_GB1410 = [ 1, 0, 2, 0, 1, 0, 2, 0, 1 ];
261 sleevePatternColors.SLV_3NTS = [ 1, 2, 3, 2, 1 ];
262 sleevePatternColors.SLV_IOWA = [ 1, 0, 2, 0, 1, 0, 2, 0, 1 ];
263 sleevePatternColors.SLV_535E = [ 1, 0, 2, 0, 1 ];
264 sleevePatternColors.SLV_NWF = [ 1, 2, 1, 0, 1, 2, 1, 0, 1, 2, 1 ];
265 sleevePatternColors.SLV_NW73 = [ 1, 0, 2, 0, 1, 0, 2, 0, 1 ];
266 sleevePatternColors.SLV_2NKNW = [ 1, 2, 1, 2, 1 ];
267 sleevePatternColors.SLV_3NKNW = [ 1, 3, 2, 3, 1 ];
268
269 /*****
270 stripePatternNames = [ "1", "2", "3E", "5E", "5NTS", "5F", "3NC", "3WC", "5WF", "5EWC", "4FF", "3NW" ];
271 insertPatternNames = [ "1", "2", "3E", "5E", "5NTS", "5F", "3WC", "5WF", "5EWC", "3NW" ];
272 braidPatternNames = [ "1", "2" ];
273
274 patternDetails = new Array();
275 patternDetails.STR_1 = [ [ 1, 1.00 ] ];
276 patternDetails.STR_2 = [ [ 1, 0.50 ], [ 2, 0.50 ] ];
277 patternDetails.STR_3E = [ [ 1, 0.33 ], [ 2, 0.33 ], [ 3, 0.33 ] ];
278 patternDetails.STR_3NC = [ [ 2, 0.40 ], [ 3, 0.20 ], [ 1, 0.40 ] ];
279 patternDetails.STR_3NW = [ [ 1, 0.25 ], [ 2, 0.25 ], [ 3, 0.50 ] ];
280 patternDetails.STR_3WC = [ [ 1, 0.20 ], [ 2, 0.60 ], [ 1, 0.20 ] ];
281 patternDetails.STR_4FF = [ [ 1, 0.20 ], [ 2, 0.20 ], [ 3, 0.20 ], [ 2, 0.40 ] ];
282 patternDetails.STR_5E = [ [ 1, 0.20 ], [ 2, 0.20 ], [ 3, 0.20 ], [ 2, 0.20 ], [ 1, 0.20 ] ];
283 patternDetails.STR_5EWC = [ [ 1, 0.15 ], [ 2, 0.15 ], [ 3, 0.40 ], [ 2, 0.15 ], [ 1, 0.15 ] ];
284 patternDetails.STR_5F = [ [ 1, 0.25 ], [ 2, 0.10 ], [ 3, 0.30 ], [ 2, 0.10 ], [ 1, 0.25 ] ];
285 patternDetails.STR_5NTS = [ [ 1, 0.20 ], [ 2, 0.25 ], [ 3, 0.10 ], [ 2, 0.25 ], [ 1, 0.20 ] ];
286 patternDetails.STR_5WF = [ [ 1, 0.35 ], [ 2, 0.10 ], [ 3, 0.10 ], [ 2, 0.10 ], [ 1, 0.35 ] ]
287
288 function patternDesc(src, thePattern)
289 {
290     var control = eval(src);
291
292     if (thePattern == NONE)
293     {
294         control.text = "NONE";
295         control.desc = "";
296         return [0,0,"NONE"];
297     }
298
299     // theStrings[0]=prefix, theString[1]=thePatternName
300     var theStrings = thePattern.split("_");
301     var theName = theStrings[1];
302     var theDesc = eval("patternDescArray."+thePattern);
```

```
303     var numStripes = theDesc[0];
304     var numColors = theDesc[1];
305
306     // lookup values in patternDescArray, else extract info from name
307     if (!isDefined(numColors))
308     {
309         numStripes = theName.charAt(0);
310         numColors = 0;
311         if (numStripes >= 3)
312         {
313             if (eval("_"+src+"Color1") == eval("_"+src+"Color3"))
314                 numColors = 2;
315             else
316                 numColors = 3;
317
318             // exceptions
319             if ((theName == "3NW") && (numColors == 2))
320                 theName = "2NW";
321             else if (theName == "3WC")
322                 numColors = 2;
323             else if (theName == "5NTS")
324                 numColors = 3;
325         }
326         else
327             numColors = numStripes;
328         theName = numColors + theName;
329     }
330
331     control.text = theName;
332     control.desc1 = numStripes + "-STRIPE / " + numColors + "-COLOR";
333     control.desc2 = numStripes + "-STRIPE" + "\n" + numColors + "-COLOR";
334
335     return [numStripes, numColors, theName];
336 }
337
338 function getPattern(src)
339 {
340     thePattern = eval("_" + src);
341     if (thePattern == null)
342         thePattern = "STR_1";
343
344     patternDesc(src, thePattern);
345
346     return thePattern;
347 }
348
349 function setObjectPattern(img, thePattern, theColors)
350 {
351     var theName = img._name;
352     if (theName.contains("Max"))
353         return;
354     //trace("setObjectPattern: img="+img+", thePattern="+thePattern+", theColors="+theColors);
355
356     if (theColors == null)
357         theColors = [ _root.NONE, _root.BLACK, _root.GREY, -1 ];
358
359     var theDetails = eval("patternDetails."+thePattern);
360     var theStripe, theColor, theScale;
361     var numStripes = theDetails.length;
362     var maxStripes = 0;
363
364     img.thePattern = thePattern;
365     img.theColors = theColors;
366
367     if (img.braidMax != null)
368     {
369         img.braidMax._visible = false;
370
371         theStripe = img.stripe0;
372         theColor = theColors[ theDetails[0][0] ];
373         setObjectColor(theStripe, theColor);
374         //trace("  braid:C1"+theColor+"-"+theStripe);
375
376         theStripe = img.stripe1;
377         theColor = (numStripes == 2) ? theColors[ theDetails[1][0] ] : NONE;
378         setObjectColor(theStripe, theColor);
379         //trace("  braid:C2"+theColor+"-"+theStripe);
380     }
381     else if (img.curveLimit != null)
382     {
383         var theWidth = img.stripe0._width;
384         var theHeight = img.stripe0._height;
385         var maxWidth = theWidth - img.curveLimit._width;
386         var maxHeight = theHeight - img.curveLimit._height;
```

```
387
388         maxStripes = 6;
389         for (i=0; i<numStripes; i++)
390         {
391             var theStripe = eval("img.stripe"+i);
392             var index = i;
393             // back of hurricanes side insert must be reversed
394             if ((_jerseyStyle == HURRICANES) && (img._name.contains("Back")))
395                 index = numStripes - i - 1;
396
397             var theColor = theColors[ theDetails[index][0] ];
398             var theScale = theDetails[index][1];
399
400             theStripe._width = theWidth;
401             theStripe._height = theHeight;
402             theWidth -= maxWidth*theScale;
403             theHeight -= maxHeight*theScale;
404             //trace("    "+i+":C"+theColor+":W="+theWidth+":H"+theHeight);
405
406             setObjectColor(theStripe, theColor);
407         }
408     }
409     else
410     {
411         var maxWidth = img.maxWidth._width;
412         var thePos = 0;
413
414         maxStripes = 12;
415         for (i=0; i<numStripes; i++)
416         {
417             var theStripe = eval("img.stripe"+i);
418             var theColor = theColors[ theDetails[i][0] ];
419             var theScale = theDetails[i][1];
420             var theWidth = maxWidth*theScale;
421
422             theStripe._width = theWidth;
423             theStripe._x = thePos;
424
425             thePos += theWidth;
426             //trace("    "+i+":C"+theColor+":W"+theWidth);
427
428             setObjectColor(theStripe, theColor);
429         }
430     }
431
432     for (; i<maxStripes; i++)
433     {
434         theStripe = eval("img.stripe"+i);
435         setObjectColor(theStripe, NONE);
436     }
437 }
438
439 function setPattern(src, thePattern)
440 {
441     patternDesc(src, thePattern);
442
443     // update uniform item's pattern
444     var img = getImage(src);
445     //trace("setPattern:"+src+": "+img+": "+thePattern);
446     if (thePattern == NONE)
447     {
448         img._visible = false;
449         return;
450     }
451     img._visible = true;
452
453     theColors = [ NONE, eval("_"+src+"Color1"), eval("_"+src+"Color2"), eval("_"+src+"Color3") ];
454
455     for (props in img)
456     {
457         theObject = eval("img."+props);
458         if (props.contains("pattern"))
459         {
460             setObjectPattern(theObject, thePattern, theColors);
461         }
462     }
463
464     if (src == "sleevePattern")
465         setValue("numTV", NONE);
466 }
467
```

```
764 function getValue(src)
765 {
766 //trace("getValue:"+src+"="+eval("_"+src));
767     return eval("_"+src);
768 }
769
770 function setValue(src,newValue)
771 {
772     oldValue = eval("_" + src);
773     eval("_"+src) = newValue;
774 //trace("setValue:"+src+"="+eval("_"+src));
775
776     if (src.endsWith("Boolean"))
777     {
778         if (src.contains("waistband"))
779         {
780             uniform.waistband.gotoAndStop(newValue);
781             setColor("beltColor",_beltColor);
782         }
783         else if (src.contains("neck"))
784         {
785             uniform.neckPattern.gotoAndStop(newValue);
786             setPattern("neckPattern",_neckPattern);
787         }
788         else if (src.contains("home"))
789         {
790             uniform.home._visible = newValue;
791         }
792     }
793     else if (src.contains("Color"))
794     {
795         setColor(src,newValue,oldValue);
796     }
797     else if (src.endsWith("Pattern"))
798     {
799         setPattern(src,newValue);
800     }
801     else if (src.endsWith("Font"))
802     {
803         setFont(src,newValue);
804     }
805     else if (src.endsWith("Text"))
806     {
807         setText(src,newValue);
808     }
809     else if (src.endsWith("Accent"))
810     {
811         // refresh updated details
812         setColor(src+"Color",eval("_"+src+"Color"));
813     }
814     else if (src == "numTV")
815     {
816         uniform.numTVImg.gotoAndStop(_numTV*FRAMEOFFSET);
817         uniform.numTVImg._visible = (_numTV != NONE);
818
819         // refresh updated details
820         setFont("numFont", _numFont)
821     }
822     else if (src == "jerseyConstruct")
823     {
824         if (_jerseyConstruct == ELITE)
825             setValue("neckBoolean",true);
826         else
827             setValue("homeBoolean",false);
828     }
829
830     else if (src == "jerseyStyleXtra")
831     {
832         toFrame = _jerseyStyleXtra*FRAMEOFFSET+FRAMEOFFSET;
833         uniform.jersey.gotoAndStop(toFrame);
834         uniform.cuffPattern.gotoAndStop(toFrame);
835         uniform.block._y = (_jerseyStyleXtra == BELT_LEN) ? -55 : -25;
836
837         if (_jerseyStyleXtra == BELT_LEN)
838             setValue("jerseyFabric", (_jerseyStyle == BATWING) ? PORTHOLE : PROBRITE);
839         else if ((_jerseyStyleXtra == CAPSLEEVE) && (_numTV == SLEEVE))
840             setValue("numTV",NONE);
841
842         // refresh updated details
843         setColor("jerseyBodyColor",_jerseyBodyColor);
844         setColor("jerseySidelColor",_jerseySidelColor);
845         setPattern("jerseySidelPattern",_jerseySidelPattern);
846         setPattern("jerseyBraid1Pattern",_jerseyBraid1Pattern);
847         setPattern("jerseyBraid2Pattern",_jerseyBraid2Pattern);
```



```
848         setPattern("cuffPattern",_cuffPattern);
849         setPattern("sleevePattern",_sleevePattern);
850     }
851     else if (src == "pantConstruct")
852     {
853         if (_pantConstruct == VARSITY)
854         {
855             setValue("pantStyle",TUNNEL);
856             setValue("pantSide1Pattern",NONE);
857             setValue("pantStyleExtra",3);
858         }
859     }
860     else if (src == "pantStyle")
861     {
862         uniform.pant.gotoAndStop(pantFrames[_pantStyle]);
863
864         if (_pantStyle == TAPERED)
865         {
866             setValue("pantConstruct",ELITE);
867             setValue("pantSide1Pattern",NONE);
868
869             setPattern("pantBraid1Pattern",_pantBraid1Pattern);
870             setPattern("pantBraid2Pattern",_pantBraid1Pattern);
871         }
872         else
873         {
874             setValue("pantBraid1Pattern",NONE);
875             setValue("pantBraid2Pattern",NONE);
876
877             setPattern("pantSide1Pattern",_pantSide1Pattern);
878         }
879
880         // refresh updated details
881         setColor("pantBodyColor",_pantBodyColor);
882         setColor("pantSide1Color",_pantSide1Color);
883     }
884     else if (src == "homeLetter")
885     {
886         var img = getImage("homeLetterColor");
887         setChar(img,_homeLetter);
888         img.outline._visible = false;
889         img.shadow._visible = false;
890     }
891     else if (src == "numFrontSize")
892     {
893         uniform.numFrontImg.num._xscale = uniform.numFrontImg.num._yscale = newValue/12*100;
894
895         // no room for text when num size = 12
896         if (newValue == 12)
897         {
898             nameTextBak = _nameText;
899             setValue("nameText","");
900         }
901         else
902         {
903             if (_nameText == "")
904                 setValue("nameText",nameTextBak);
905         }
906
907         // refresh updated details
908         setValue("nameSize",_nameSize);
909     }
910 }
911 else if (src == "nameSize")
912 {
913     uniform.nameImg._xscale = uniform.nameImg._yscale = newValue/3*100;
914
915     var thePoint = new Array();
916     var target = (_numFrontSize == 12) ? uniform.nameImg.top : uniform.nameImg.bottom;
917     thePoint.x = target._x;
918     thePoint.y = target._y;
919     uniform.nameImg.localToGlobal(thePoint);
920     uniform.globalToLocal(thePoint);
921     uniform.numFrontImg._y = thePoint.y;
922 }
923
924 if (uniform == previewUniform)
925 {
926     if (isZoomed)
927     {
928         uniform = zoomed.uniform;
929         setValue(src,newValue);
930         uniform = previewUniform;
931     }
```

```
932
933         notifyListeners(src);
934         calculatePrice();
935     }
936 }
937
938 function hasVarsity()
939 {
940     return (_jerseyStyle == LINEMAN) || (_jerseyStyle == SKILL) || (_jerseyStyle == MESH);
941 }
942
943 /*****
944 // should rename to initControl
945 function getEnabled(target)
946 {
947     target.stop();
948     var src = target._name;
949 //trace("getEnabled:"+src+":"+target);
950
951     if (src.endsWith("None"))
952     {
953         // colorNone (must precede Color)
954         target.update = function() { with (_root)
955         {
956             this.theName = this._name.substring(0, this._name.indexOf("None"));
957             var isEnabled = (getValue(this.theName) == NONE);
958             setEnabled(this,isEnabled);
959             eval(this.theName).hilite._visible = !isEnabled;
960
961             // dynamic details
962             teamColor3None._visible = (_teamColor4 == NONE);
963             cuffPatternNone._visible = (_jerseyConstruct == VARSITY);
964             neckPatternNone._visible = false;
965
966             }; }; target.update();
967         }
968     else if (src.endsWith("Boolean"))
969     {
970         // varBoolean (must precede others with simialr prefix, i.e. home)
971         target.update = function() { with (_root)
972         {
973             var src = this._name;
974
975             this.theValue = getValue(src);
976             this.isEnabled = true;
977             if (this == neckBoolean)
978                 this.isEnabled = (_jerseyConstruct == VARSITY);
979             else if (this == homeBoolean)
980                 this.isEnabled = ((_jerseyConstruct == ELITE) && (_jerseyStyle != HURRICANES));
981
982 //trace("getEnabled boolean:"+src+":"+this.isEnabled)
983             if (this.isEnabled)
984                 this.gotoAndStop(this.theValue)
985             else
986                 this.gotoAndStop("off");
987
988             }; }; target.update();
989         }
990     else if (src.startsWith("teamColor"))
991     {
992         // teamColors1..4 (must precede Color)
993         target.update = function()
994         {
995             var i, theIndex = this._name.charAt(this._name.length-1)-1;
996             var theLength = _root.teamColors.length;
997
998             if (this == _root.teamColor4)
999             {
1000                 isEnabled = _root._teamColor3 != _root.NONE;
1001                 _root.setEnabled(_root.teamColor4, isEnabled);
1002                 _root.teamColor4None._visible = isEnabled;
1003             }
1004             else
1005                 _root.setEnabled(this,true);
1006
1007             theOptions = _root.allColorOptions;
1008             for (i=0; i<theLength; i++)
1009             {
1010                 if (i != theIndex)
1011                     theOptions = _root.removeItem(theOptions,_root.teamColors[i]);
1012             }
1013             this.setOptions(theOptions);
1014
1015             }; target.update();
```

```
1016     }
1017     else if (src.endsWith("AccentColor"))
1018     {
1019         // num/nameAccentColor1..3
1020         target.update = function()
1021         {
1022             var src = this._name;
1023             var prefix = src.substring(0, src.indexOf("Accent"));
1024             var theAccent = _root.getValue(prefix+"Accent");
1025             var isEnabled = (theAccent != _root.NONE);
1026
1027             if (isEnabled)
1028             {
1029                 theOptions = _root.teamColors;
1030                 theOptions = _root.removeItem(theOptions, _root._jerseyBodyColor);
1031                 theOptions = _root.removeItem(theOptions, eval("_root._"+prefix+"MainColor"));
1032
1033                 label = (theAccent == _root.SHADOW) ? "Shadow" :
1034                     ((_root.getValue(prefix+"Font") == _root.INLINE) ? "Inline" : "Outline");
1035                 label += " Color";
1036             }
1037             _root.setEnabled(this, isEnabled);
1038
1039             }; target.update();
1040     }
1041     else if (isDefined(getValue(target._parent._name)))
1042     {
1043         // toggle buttons embedded in var control
1044         target.update = function()
1045         {
1046             var src = this._name;
1047             var theVar = _parent._name;
1048             var theFadeName = "_parent."+src+"Fade";
1049             var theFade = eval(theFadeName);
1050
1051             // isEnabled & theValue referenced elsewhere, must persist beyond func
1052             theValue = (_name.charAt(0) == "_" ? parseInt(_name.substring(1)) : eval("_root."+_name);
1053             if (!_root.isDefined(isEnabled))
1054                 isEnabled = true;
1055
1056             if (isEnabled)
1057                 this.gotoAndStop(this.theValue == _parent.theValue);
1058             else
1059             {
1060                 this.gotoAndStop("off");
1061
1062                 if (theFade == null)
1063                 {
1064                     _parent.attachMovie("fade", src+"Fade", theValue);
1065                     theFade = eval(theFadeName);
1066                     theFade._x = this._x;
1067                     theFade._y = this._y;
1068                     theFade._width = this._width + 4;
1069                     theFade._height = this._height + 4;
1070                     theFade._alpha = 70;
1071                     theFade._visible = false;
1072                 }
1073             }
1074
1075             theFade._visible = !isEnabled;
1076
1077             }; target.update();
1078
1079         target.excludeOthers = function ()
1080         {
1081             for (prop in _parent)
1082             {
1083                 button = eval("_parent."+prop);
1084                 if ((button != this) && (button.theValue != null) && (button.isEnabled))
1085                     button.gotoAndStop(false);
1086             }
1087         }
1088     }
1089     else if (src.contains("numTV"))
1090     {
1091         if ((_jerseyStyle == HURRICANES) || (_jerseyStyleExtra == CAPSLEEVE) || (_sleevePattern != NONE))
1092             target.SLEEVE.isEnabled = false;
1093
1094         if ((_jerseyStyle == TERPS) || (_jerseyStyle == USC) || (_jerseyStyle == DUCKS) || (_jerseyStyle ==
1095             SHOULDDER_INS))
1096             target.SHOULDER.isEnabled = false;
1097
1098         target.theValue = _root.getValue(src);
1099         setEnabled(target, true);
1100     }
```

```
1099     }
1100     else if (src.endsWith("Accent"))
1101     {
1102         var prefix = src.substring(0,src.indexOf("Accent"));
1103         var theFont = getValue(prefix+"Font");
1104
1105         if (theFont >= INLINE)
1106             target.SHADOW.isEnabled = false;
1107
1108         if (theFont == INLINE)
1109             target.outlineLabel = "INLINE";
1110
1111         target.theValue = _root.getValue(src);
1112         setEnabled(target,true);
1113     }
1114     else if (src.endsWith("App"))
1115     {
1116         var prefix = src.substring(0,src.indexOf("App"));
1117         var theFont = getValue(prefix+"Font");
1118         var isEnabled = (theFont != INLINE) && (theFont != DOUBLE);
1119
1120         target.theValue = _root.getValue(src);
1121         setEnabled(target,isEnabled);
1122     }
1123     else if (src.contains("PatternColor"))
1124     {
1125         //varPatternColor1..3
1126         target.update = function()
1127         {
1128             var src = this._name;
1129             var theIndex = src.charAt(src.length-1);
1130             var theVar = src.substring(0,src.indexOf("Color"));
1131             var thePattern = _root.getValue(theVar);
1132             var maxColors = eval("_root.patternDescArray."+thePattern)[1];
1133
1134             this.label = "Color "+theIndex;
1135
1136             // for STR_patterns, this is numStripes but must allow for 2-3 colors
1137             if (thePattern == "STR_3WC")
1138                 maxColors = 2;
1139             else if (!_root.isDefined(maxColors))
1140                 maxColors = thePattern.charAt(4);
1141
1142             var isEnabled = (theIndex <= maxColors);
1143             if (isEnabled)
1144             {
1145                 // theOptions property used by setOptions
1146                 this.theOptions = _root.teamColors;
1147                 if (theIndex == 2)
1148                     theOptions = _root.removeItem(theOptions,eval("_root."+theVar+"Color1"));
1149                 else if (theIndex == 3)
1150                 {
1151                     theOptions = _root.removeItem(theOptions,eval("_root."+theVar+"Color2"));
1152                     if (thePattern.contains("SLV"))
1153                         theOptions = _root.removeItem(theOptions,eval("_root."+theVar+"Col
1154 or1"));
1155                 }
1156                 _root.setEnabled(this,isEnabled);
1157             }
1158             target.update();
1159         }
1160     else if (src.endsWith("Pattern"))
1161     {
1162         target.update = function()
1163         {
1164             var src = this._name;
1165             var isEnabled = true;
1166
1167             this.theValue = _root.getValue(src);
1168             var thePrefix = "_root." + src;
1169             var theColors = [_root._jerseyBodyColor, eval(thePrefix+"Color1"), eval(thePrefix+"Color2")
, eval(thePrefix+"Color3")];
1170             _root.setObjectPattern(hilite.patternBox, this.theValue, theColors);
1171
1172             _root.setEnabled(this,isEnabled);
1173         };
1174         target.theOptions = eval(target.theOptions);
1175         target.setOptions = patternSetOptions;
1176         target.update();
1177     }
1178     else if (src.startsWith("home"))
1179     {
1180         // homeBody/LetterColor (not homeBoolean which is handled by Booleans general
```

```
1181         target.update = function()
1182         {
1183             this.theOptions = _root.teamColors;
1184             _root.setEnabled(this, _root._homeBoolean == true);
1185
1186             if ((this == _root.homeBodyColor) && (_root._jerseyStyle == _root.HURRICANES))
1187                 _root.setEnabled(this, false);
1188
1189             }; target.update();
1190         }
1191     else if (src.contains("Color"))
1192     {
1193         // all colors not specified above (i.e. accent & patterns)
1194         target.update = function() { with (_root)
1195         {
1196             var src = this._name;
1197             var isEnabled = (getImage(src) != null) || (!isDefined(eval("varMap."+src)));
1198             setEnabled(this, isEnabled);
1199
1200             }; }; target.update();
1201         }
1202     else if (src.endsWith("Construct"))
1203     {
1204         target.update = function() { with (_root)
1205         {
1206             // thisOptions used within control
1207             this.theOptions = ["ELITE", "VARSITY"];
1208
1209             var src = this._name;
1210             var theValue = getValue(src);
1211             var isEnabled = src.startsWith("jersey") ? hasVarsity() : (_pantStyle == TUNNEL);
1212
1213             if (isEnabled)
1214                 this.gotoAndStop(theOptions[theValue]);
1215             else
1216                 this.gotoAndStop("off");
1217
1218             }; }; target.update();
1219         }
1220     else if (src.startsWith("jerseyStyleExtra"))
1221     {
1222         target.theValue = _root.getValue(src);
1223
1224         if (_jerseyStyle == MESH)
1225             target.gotoAndStop("off");
1226         else if ((_jerseyStyle == BATWING) || (_jerseyStyle == LINEMAN))
1227             target.gotoAndStop("NONE");
1228         else if ((_jerseyStyle == SKILL) || (_jerseyStyle == SHOULDER_INS))
1229             target.gotoAndStop("BELT_LEN");
1230         else
1231             target.gotoAndStop("CAPSLEEVE");
1232     }
1233     else if (src.startsWith("jerseyFabric"))
1234     {
1235         target.update = function() { with (_root)
1236         {
1237             var src = this._name;
1238             this.theValue = getValue(src);
1239             var isEnabled = true;
1240
1241             if ((_jerseyStyle == SKILL) && (_jerseyConstruct == VARSITY))
1242                 void(true);
1243             else if ((_jerseyStyleExtra == BELT_LEN) || (_jerseyStyle == MESH))
1244                 isEnabled = false;
1245
1246             if (isEnabled)
1247                 this.gotoAndStop(1);
1248             else
1249                 this.gotoAndStop(this.theValue*FRAMEOFFSET+FRAMEOFFSET);
1250
1251             }; }; target.update();
1252         }
1253     else if (src.startsWith("pantStyleExtra"))
1254     {
1255         target.update = function() { with (_root)
1256         {
1257             var isEnabled = (_pantStyle == TUNNEL) && (_pantConstruct == ELITE);
1258             this.theValue = getValue(this._name);
1259             setEnabled(this, isEnabled);
1260
1261             }; }; target.update();
1262         }
1263     else
1264     {
```

```
1265         target.theValue = _root.getValue(src);
1266         setEnabled(target,true);
1267     }
1268 }
1269
1270 function setEnabled(target,isEnabled)
1271 {
1272     target.isEnabled = isEnabled;
1273     if (target._name.contains("Boolean"))
1274         target.gotoAndStop(isEnabled);
1275     else
1276         target.gotoAndStop(isEnabled?"on":"off");
1277     target.setOptions(null);
1278 }
```

```
1456 function notifyListeners(theProperty)
1457 {
1458     var theControls = eval("listeners."+theProperty);
1459     if (theControls == null) return;
1460
1461     //trace("notifyListeners of "+theProperty+": "+theControls)
1462     for (i=0; i<theControls.length; i++)
1463     {
1464         //trace("    "+theControls[i]+".update");
1465         eval(theControls[i]).update();
1466     }
1467 }
1468
1469 function initDefaults(teamColor1, teamColor2, teamColor3, teamColor4)
1470 {
1471     // defaults not spec'd in template, careful not to trample
1472     _jerseyStyleExtra = NONE;
1473     _jerseyConstruct = ELITE;
1474     _jerseyFabric = PORTHOLE;
1475
1476     _pantStyle = TUNNEL;
1477     _pantStyleExtra = 3;
1478     _pantConstruct = _jerseyConstruct;
1479
1480     _neckBoolean = true;
1481     _homeBoolean = false;
1482     _waistbandBoolean = false;
1483
1484     _jerseyBodyColor = teamColor1;
1485     _jerseySide1Color = teamColor2;
1486
1487     _pantBodyColor = teamColor1;
1488     _pantSide1Color = teamColor2;
1489     _beltColor = teamColor2;
1490
1491     _jerseySide1Pattern = _jerseyBraid1Pattern = _jerseyBraid2Pattern = _pantSide1Pattern = _pantBraid1Pattern
= _pantBraid2Pattern = _sleevePattern = NONE;
1492     _jerseySide1PatternColor1 = _pantSide1PatternColor1 = _jerseyBraid1PatternColor1 = _jerseyBraid2PatternColor1
= _pantBraid1PatternColor1 = _pantBraid2PatternColor1 = _sleevePatternColor1 = teamColor3;
1493     _jerseySide1PatternColor2 = _pantSide1PatternColor2 = _jerseyBraid1PatternColor2 = _jerseyBraid2PatternColor2
= _pantBraid1PatternColor2 = _pantBraid2PatternColor2 = _sleevePatternColor2 = teamColor2;
1494     _jerseySide1PatternColor3 = _pantSide1PatternColor3 = teamColor3;
1495     _sleevePatternColor3 = teamColor1;
1496
1497     _neckPattern = _cuffPattern = "STR_1";
1498     _neckPatternColor1 = _cuffPatternColor1 = teamColor2;
1499     _neckPatternColor2 = _cuffPatternColor2 = teamColor3;
1500     _neckPatternColor3 = _cuffPatternColor3 = teamColor2;
1501
1502     _numFont = FULL;
1503     _numFontSize = 10;
1504     _numApp = TWILL;
1505     _numMainColor = teamColor2;
1506     _numAccentColor = teamColor3;
1507     _numAccent = NONE;
1508     _numTV = SLEEVE;
1509
1510     _nameFont = FULL;
1511     _nameSize = 3;
1512     _nameApp = TWILL;
1513     _nameText = "TEAM NAME";
1514     _nameMainColor = teamColor2;
1515     _nameAccentColor = teamColor3;
1516     _nameAccent = NONE;
1517
1518     _homeBodyColor = teamColor1;
1519     _homeLetterColor = teamColor2;
1520     _homeLetter = "A";
1521
1522     // hack
1523     if (teamColor1 == WHITE)
1524     {
1525         _pantBodyColor = teamColor2;
1526         _pantSide1Color = teamColor1;
1527         _pantSide1PatternColor1 = teamColor1;
1528         _beltColor = teamColor1;
1529     }
1530     _neckPatternColor1 = _cuffPatternColor1 = _beltColor = WHITE;
1531 }
1532
1533 function initBuilder()
1534 {
1535     // main preview, uniform variable toggled for zoom
1536     uniform = previewUniform;
```

```
1537
1538     previewColorItems = ["jerseyBodyColor", "jerseySide1Color", "pantBodyColor", "pantSide1Color", "beltColor"]
1539 ;
1540 // list of uniform objects that need colorization based on global var containing word "color"
1541 // exclude teamColor# to avoid indefinite loop
1542 colorItems = new Array();
1543 for (globalVar in _root)
1544 {
1545     if (globalVar.charAt(0)=='_')
1546     {
1547         eval("listeners."+globalVar.substring(1)) = new Array();
1548         if ((globalVar.indexOf("Color") > 0) && (globalVar.indexOf("team")<0))
1549             colorItems.push(globalVar.substring(1));
1550     }
1551 }
1552
1553 // re-org stripe pattern names
1554 theArray = stripePatternNames;
1555 for (i=0; i<theArray.length; i++)
1556     theArray[i] = "STR_"+theArray[i];
1557
1558 theArray = insertPatternNames;
1559 for (i=0; i<theArray.length; i++)
1560     theArray[i] = "STR_"+theArray[i];
1561
1562 theArray = braidPatternNames;
1563 for (i=0; i<theArray.length; i++)
1564     theArray[i] = "STR_"+theArray[i];
1565
1566 theArray = sleevePatternNames;
1567 patternDescArray = new Array();
1568 for (i=0; i<theArray.length; i++)
1569 {
1570     theArray[i] = thePattern = "SLV_"+theArray[i];
1571
1572     var theWidths = eval("sleevePatternWidths."+thePattern);
1573     var theColors = eval("sleevePatternColors."+thePattern);
1574     var totalStripes = theWidths.length;
1575     var sumWidths = 0;
1576
1577     var theDetails = new Array();
1578     var numColors = 0;
1579     var numStripes = totalStripes;
1580     for (j=0; j<totalStripes; j++)
1581     {
1582         sumWidths += theWidths[j];
1583         theColor = theColors[j];
1584         theDetails[j] = [ theColor, theWidths[j] ];
1585
1586         numColors = Math.max(numColors, theColor);
1587         if (theColor == 0) numStripes--;
1588     }
1589     eval("patternDetails."+thePattern) = theDetails;
1590     eval("patternDescArray."+thePattern) = [ numStripes, numColors ];
1591
1592     // center narrow patterns, else scale to fit width
1593     if (sumWidths/SLV_WIDTH < .8)
1594     {
1595         for (j=totalStripes; j>0; j--)
1596         {
1597             theDetails[j] = theDetails[j-1];
1598             theDetails[j][1] /= SLV_WIDTH;
1599         }
1600         pad = [ 0, (SLV_WIDTH-sumWidths)*.5/SLV_WIDTH ];
1601         theDetails[0] = pad;
1602         theDetails[theDetails.length] = pad;
1603     }
1604     else
1605     {
1606         for (j=0; j<totalStripes; j++)
1607         {
1608             theDetails[j][1] /= sumWidths;
1609         }
1610     }
1611 }
1612
1613 // 1st go to start page to init nav funcs
1614 gotoAndStop("start");
1615 if (notSavedUniforms.length>0)
1616 {
1617     initUser(notSavedUniforms[0]);
1618     if (CusNum == -1)
1619     {
```



```
1620                // 1st go to print page to init vars
1621                // then go to dealer view with disabled non-dealer steps & buttons
1622                gotoAndStop("print");
1623                gotoAndStop("dealer");
1624            }
1625            else
1626            {
1627                gotoAndStop("colors");
1628            }
1629        }
1630
1631        // color mapping variables to uniform graphic symbol names
1632        varMap = new Array();
1633        varMap.jerseyBodyColor = "jersey.jerseyBodyColor";
1634        varMap.jerseySide1Color = "jersey.jerseySide1Color";
1635        varMap.jerseySide1Pattern = "jersey.jerseySide1Pattern";
1636        varMap.jerseyBraid1Pattern = "jersey.jerseyBraid1Pattern";
1637        varMap.jerseyBraid2Pattern = "jersey.jerseyBraid2Pattern";
1638
1639        varMap.pantBodyColor = "pant.pantBodyColor";
1640        varMap.pantSide1Color = "pant.pantSide1Color";
1641        varMap.pantSide1Pattern = "pant.pantSide1Pattern";
1642        varMap.pantBraid1Pattern = "pant.pantBraid1Pattern";
1643        varMap.pantBraid2Pattern = "pant.pantBraid2Pattern";
1644
1645        varMap.beltColor = "waistband.beltColor";
1646        varMap.neckPattern = "neckPattern";
1647        varMap.cuffPattern = "cuffPattern";
1648        varMap.sleevePattern = "jersey.sleevePattern";
1649
1650        varMap.homeBodyColor = "home.homeBodyColor";
1651        varMap.homeLetterColor = "home.homeLetterColor";
1652    }
1653
1654    //initialize setup based on defaults
1655    var _teamColor1, _teamColor2, _teamColor3, _teamColor4;
1656
1657    var _jerseyStyle, _jerseyStyleExtra, _jerseyConstruct, _jerseyFabric, _neckBoolean, _homeBoolean;
1658    var _pantStyle, _pantStyleExtra, _pantConstruct, _waistbandBoolean;
1659    var _jerseyBodyColor, _jerseySide1Color, _pantBodyColor, _pantSide1Color, _beltColor;
1660
1661    var _jerseyBraid1Pattern, _jerseyBraid1PatternColor1, _jerseyBraid1PatternColor2;
1662    var _jerseyBraid2Pattern, _jerseyBraid2PatternColor1, _jerseyBraid2PatternColor2;
1663    var _pantBraid1Pattern, _pantBraid1PatternColor1, _pantBraid1PatternColor2;
1664    var _pantBraid2Pattern, _pantBraid2PatternColor1, _pantBraid2PatternColor2;
1665
1666    var _jerseySide1Pattern, _jerseySide1PatternColor1, _jerseySide1PatternColor2, _jerseySide1PatternColor3;
1667    var _pantSide1Pattern, _pantSide1PatternColor1, _pantSide1PatternColor2, _pantSide1PatternColor3;
1668    var _sleevePattern, _sleevePatternColor1, _sleevePatternColor2, _sleevePatternColor3;
1669    var _neckPattern, _neckPatternColor1, _neckPatternColor2, _neckPatternColor3;
1670    var _cuffPattern, _cuffPatternColor1, _cuffPatternColor2, _cuffPatternColor3;
1671
1672    var _numFont, _numFrontSize, _numApp, _numTV, _numMainColor, _numAccentColor, _numAccent;
1673    var _nameFont, _nameSize, _nameApp, _nameText, _nameMainColor, _nameAccentColor, _nameAccent;
1674    var _homeBodyColor, _homeLetterColor, _homeLetter;
1675
1676    var _UID, _saveName, _saveSchool, _saveCity, _saveState;
1677
1678
1679    // prices & style codes stored externally
1680    #include "football-custom.as"
1681
1682    initBuilder();
```