

```
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, delete.asp, save.asp, urls.txt
12 //
13 // FILES REQUIRED @ GENTIME:
14 // CenterCourtMen.as or CenterCourtWomen.as
15 //
16 // Pricing and code specific to generating men's or women's version of the
17 // builder is stored externally in the .as files mentioned above.
18 // The library folder "_gender-details" contains symbols for men/women.
19 // The frame("script") contains documentation on how to generate men/women
20 // versions of each builder.
21 //
22 /*****/
23
24 // load urls for button links: membersLink, dealersLink, builderLink
25 loadVariables("urls.txt", this);
26
27 user = "";
28 password = "";
29
30 timeout = 30000;
31 start = getTimer();
32 index = -1;
33 xmlReady = false;
34 savedUniforms = null;
35 totalBytes = _root.getBytesTotal();
36 savedUniforms = new Array();
37 uniformName = "";
38 msg = "";
39
40 CODE0 = "0".charCodeAt(0);
41 CODE9 = "9".charCodeAt(0);
42 function getUniform(node)
43 {
44     var theUniform = node.attributes;
45     for (props in theUniform)
46     {
47         var theName = "theUniform."+props;
48         var theValue = eval(theName);
49         if ((theValue.charCodeAt(0) >= CODE0) && (theValue.charCodeAt(0) <= CODE9))
50             eval(theName) = parseInt(theValue);
51         else if (theValue == "true")
52             eval(theName) = true;
53         else if (theValue == "false")
54             eval(theName) = false;
55     }
56
57     return theUniform;
58 }
59
60 function getSavedList()
61 {
62     if ((!this.loaded) || (this.status != 0))
63     {
64         // error in xml, move on w/o saved list
65         xmlReady = true;
66         return;
67     }
68
69     var top = this.lastChild;
70     while (top.nodeName != "savedList")
71     {
72         top = top.previousSibling;
73     }
74
75     top = top.firstChild;
76     i=0;
77     while (top.nodeName == "uniform")
78     {
79         var theUniform = getUniform(top);
80         theUniform.index = i;
81         savedList.addItem(theUniform);
82         _root.savedUniforms.push(theUniform);
83         top = top.nextSibling;
84         i++;
85     }
86 }
```

```
85     }
86
87     xmlReady = true;
88 }
89
90 // if loading from another builder, do not redisplay login screen
91 if (eval("cusNum") != null)
92 {
93     _cusNum = parseInt(cusNum);
94     if (_cusNum >= 0)
95     {
96         thexml = new XML();
97         thexml.onLoad = getSavedList;
98         thexml.load("load.asp");
99     }
100 else
101 {
102     xmlReady = true;
103 }
104
105 start = getTimer();
106
107 // wait until login screen loaded to bypass
108 stop();
109 }
110 else
111 {
112     // move onto login screen
113     play();
114 }
```

```
1 /*****/
2 //
3 // GENERATING MENS/WOMENS SPECIFIC BUILDER:
4 //
5 // 1. Comment/Uncomment men/women .as script include below
6 // 2. Swap instance in frames
7 //   a. 25 script - banner, uniform
8 //   b. 30 start - template, templateList
9 //   c. 40 body - zoomed.uniform (maybe hidden behind help)
10 //   Hint: check the use count to ensure only correct symbols are used
11 // 3. Generate .swf then rename to appropriate
12 //
13 /*****/
14
15 // #include "CenterCourtMen.as"
16 #include "CenterCourtWomen.as"
17
18 // color constants
19 NONE = 0;
20 WHITE = 1;
21 BLACK = 2;
22 BROWN = 3;
23 NAVY = 4;
24 PURPLE = 5;
25 ROYAL = 6;
26 POWDER = 7;
27 GREEN = 8;
28 KELLY = 9;
29 MAROON = 10;
30 CARDINAL = 11;
31 SCARLET = 12;
32 ORANGE = 13;
33 GOLD = 14;
34 VEGAS = 15;
35 GREY = 16;
36
37 // font constants
38 GOTHIC = 1;
39 FULL = 2;
40 NIKEBLOCK = 3;
41 INLINE = 4;
42 DOUBLE = 5;
43
44 // style constants
45 STRAIGHT = 1;
46 ROTATE = 2;
47 SKEWED = 3;
48 BOTTOM = 4;
49
50 // application constants
51 TWILL = 1;
52 PRINT = 2;
53
54 // fabric constants
55 SHEEN = 0;
56 MESH = 1;
57 BRITE = 2;
58
59 // waistband constants
60 KNIT = 0;
61 NEEDLE = 1;
62
63 // accent constants
64 OUTLINE = 1;
65 SHADOW = 2;
66
67 MAXCHARS = 13;
68 FRAMEOFFSET = 5;
69 CODEA = "A".charCodeAt(0);
70 CODEZ = "Z".charCodeAt(0);
71
72 fabricStyles = new Array(
73     "SHIMMERSHEEN", "MICROMESH", "PRO-BRITE"
74 );
75
76 nameStyles = new Array(
77     "none", "STRAIGHT", "ARCHED", "VERTICAL ARCHED", "TOP & BOTTOM"
78 );
79
80 fontNames = new Array(
81     "none",
82     "GOTHIC",
83     "FULL BLOCK",
84     "NIKE TRUE BLOCK",
```

```
85     "NIKE INLINE",
86     "DOUBLE BLOCK"
87 );
88
89 fontDesc = new Array(
90     "none",
91     "GOTHIC Solid/Outline/Shadow",
92     "FULL BLOCK Solid/Outline/Shadow",
93     "NIKE TRUE BLOCK Solid/Outline/Shadow",
94     "NIKE INLINE Solid/Inline\nAvailable Printed Only",
95     "DOUBLE BLOCK Solid/Outline\nAvailable Printed Only"
96 );
97
98 fontName = new Array(
99     "none",
100    "gothic", "full", "nikeblock", "inline", "double"
101 );
102
103 colorText = new Array(
104     "NONE",
105     "WHITE", "BLACK", "BROWN", "NAVY", "PURPLE", "ROYAL", "POWDER BLUE", "DARK GREEN",
106     "KELLY", "MAROON", "CARDINAL", "SCARLET", "ORANGE", "GOLD", "VEGAS GOLD", "GREY"
107 );
108
109 colorHex = new Array(
110     -1,
111     0xfffff, 0x000000, 0x330000, 0x000066,
112     0x330066, 0x003399, 0x6699CC, 0x006600,
113     0x006633, 0x660033, 0x990000, 0xCC0000,
114     0xFF6600, 0xFFCC33, 0CCCC99, 0x999999
115 );
116
117 colorItems = new Array(
118     "bodyColor", "side1Color", "side2Color", "side3Color", "trim1Color", "trim2Color", "trim3Color",
119     "numMain", "numAccentColor", "nameMain", "nameAccentColor", "homePlateColor"
120 );
121
122 function countLetters(theName)
123 {
124     var numLetters = 0;
125     numLetters = theName.length;
126     for (var i=numLetters-1; i>=0; i--)
127     {
128         if (theName.charAt(i)==' ')
129             numLetters--;
130     }
131     return numLetters;
132 }
133
134 function markupPrice(theNumber)
135 {
136     return Math.round(theNumber*20125)/10000;
137 }
138
139 function formatPrice(theNumber)
140 {
141     theNumber = Math.round(theNumber*100);
142     if ((theNumber%100) == 0)
143         return "$"+theNumber/100+".00";
144     else if ((theNumber%10) == 0)
145         return "$"+theNumber/100+"0";
146     else
147         return "$"+theNumber/100;
148 }
149
150 function calculatePrice()
151 {
152     logosPrice = _homePlate ? homePlatePrice : 0;
153     jerseyPrice = garmentPrice[_uniformStyle][_fabric];
154     if (hasWaistband())
155         shortsPrice = garmentPrice[_uniformStyle][_fabric+3*( _waistband+1)];
156     else
157         shortsPrice = garmentPrice[_uniformStyle][_fabric+3];
158
159     // print: 2 loc front + back
160     // twill: 4 numbers = 2 front + 2 back
161     if ( _numApp == PRINT)
162         numPrice = numPrintPrice[( _numFont>=INLINE)?1:0][( _numAccent==NONE)?0:1] * 2;
163     else
164         numPrice = numTwillPrice[_numFrontSize][_numAccent]*2 + numTwillPrice[_numBackSize][_numAccent]*2;
165
166     // print: top + bottom
167     // twill: num letters
168     if ( _nameApp == PRINT)
```

```
169         namePrice = namePrintPrice[(_nameAccent==NONE)?0:1] * ((_nameStyle == BOTTOM)?2:1);
170     else
171     {
172         numLetters = countLetters(_nameTop) + ((_nameStyle == BOTTOM)?countLetters(_nameBottom):0);
173         namePrice = nameWillPrice[_nameSize][_nameAccent] * numLetters;
174     }
175
176     // markup
177     jerseyPrice = markupPrice(jerseyPrice);
178     shortsPrice = markupPrice(shortsPrice);
179     numPrice = markupPrice(numPrice);
180     namePrice = markupPrice(namePrice);
181     logosPrice = markupPrice(logosPrice);
182     totalPrice = jerseyPrice + shortsPrice + numPrice + namePrice + logosPrice;
183
184     jerseyPrice = formatPrice(jerseyPrice);
185     shortsPrice = formatPrice(shortsPrice);
186     numPrice = formatPrice(numPrice);
187     namePrice = formatPrice(namePrice);
188     logosPrice = formatPrice(logosPrice);
189     totalPrice = formatPrice(totalPrice);
190 }
191
192 function getColor(src)
193 {
194     theColor = eval("_" + src);
195     if (theColor == null)
196         theColor = 0;
197     eval(src + "Text") = colorText[theColor];
198     return theColor;
199 }
200
201 function setObjectColor(img, theColor)
202 {
203     img._visible = (theColor == NONE) ? false : true;
204     // deselect none and set item's color
205     var newColor = new Color(img);
206     if (theColor > 0)
207         newColor.setRGB(colorHex[theColor]);
208     else
209         newColor.setRGB(0x000000);
210 }
211
212 function replaceColor(oldColor, newColor)
213 {
214     for (var i=0; i<colorItems.length; i++)
215         if ((eval("_"+colorItems[i]) == oldColor) && (oldColor != NONE))
216             setColor(colorItems[i],newColor);
217 }
218
219 function setTeamColor(src, theColor)
220 {
221     // update global variable (same as object name)
222     var oldColor = eval("_" + src);
223     eval("_" + src) = theColor;
224
225     // update description, none, and color
226     eval(src + "Text") = colorText[theColor];
227     if (theColor == NONE)
228     {
229         eval(src).hilite._visible = false;
230     }
231     else
232     {
233         eval(src).hilite._visible = true;
234         eval(src+"None").gotoAndStop("off");
235     }
236
237     // update variables
238     var theNum = src.charAt(9);
239     var isDuplicate = false;
240     for (var i=1; i<=4; i++)
241     {
242         if ((eval("_teamColor"+i) == theColor) && (i != theNum) && (theColor != NONE))
243         {
244             isDuplicate = true;
245         }
246     }
247
248     if (!isDuplicate)
249     {
250         teamColors = new Array(0,0,0,0);
251         teamColorMsg = "";
252         replaceColor(oldColor,theColor);
```

```
253
254     var i=0;
255     // consolidate duplicates and none selected
256     for (var j=0; j<4; j++)
257     {
258         var theColor = eval("_teamColor"+(j+1));
259         if (theColor != NONE)
260         {
261             teamColors[i] = theColor;
262             i++;
263         }
264     }
265
266     // update visuals
267     for (var j=0; j<4; j++)
268     {
269         var theColor = teamColors[j];
270         var button = selection.buttonArray[j];
271         button.theColor = theColor;
272         setObjectColor(button,teamColors[j]);
273         selection.frameArray[j]._visible = (theColor != NONE);
274     }
275     return true;
276 }
277 else
278 {
279     teamColorMsg = "Sorry, duplicate colors not allowed\nwhen selecting team colors.";
280     eval("_" + src) = oldColor;
281     eval(src + "Text") = colorText[oldColor];
282     if (oldColor == NONE)
283     {
284         eval(src).hilite._visible = false;
285         eval(src+"None").gotoAndStop("on");
286     }
287     else
288     {
289         eval(src).hilite._visible = true;
290     }
291     return false;
292 }
293 }
294
295 function setColor(src, theColor)
296 {
297     // update global variable (same as object name)
298     eval("_" + src) = theColor;
299
300     // update description, none, and color
301     if (theColor == NONE)
302     {
303         eval(src).hilite._visible = false;
304     }
305     else
306     {
307         eval(src).hilite._visible = true;
308         eval(src+"None").gotoAndStop("off");
309     }
310
311     if (src.indexOf("Accent")>0)
312     {
313         if (src.indexOf("num")==0)
314         {
315             prefix = "num";
316             theAccent = _numAccent;
317         }
318         else
319         {
320             prefix = "name";
321             theAccent = _nameAccent;
322         }
323
324         if (theAccent==OUTLINE)
325         {
326             setColor(prefix+"Outline",theColor);
327             setColor(prefix+"Shadow",NONE);
328         }
329         else if (theAccent==SHADOW)
330         {
331             setColor(prefix+"Outline",NONE);
332             setColor(prefix+"Shadow",theColor);
333         }
334         else
335         {
336             setColor(prefix+"Outline",NONE);
```

```
337         setColor(prefix+"Shadow",NONE);
338     }
339 }
340 else if (src.indexOf("num") == 0)
341 {
342     // layer = main, outline, or shadow
343     var layer = src.substr(3);
344
345     // update font + back numbers color + visible
346     setObjectColor(eval("numFrontImg." + layer),theColor);
347     setObjectColor(eval("numBackImg." + layer),theColor);
348 }
349 else if (src.indexOf("name") == 0)
350 {
351     // layer = main, outline, or shadow
352     var layer = src.substr(4);
353
354     // update font numbers color + visible
355     for (var i=1; i<=MAXCHARS; i++)
356     {
357         setObjectColor(eval("nameImg.top.char" + i + "." + layer),theColor);
358         setObjectColor(eval("nameImg.bottom.char" + i + "." + layer),theColor);
359     }
360 }
361 else if (src.indexOf("home") == 0)
362 {
363     setObjectColor(uniform.homePlateShow.letter,theColor);
364 }
365 else
366 {
367     // update uniform item's color
368     setObjectColor(eval("uniform." + src + "Img"),theColor);
369
370     // update swoosh to contrast with body
371     if (src == "bodyColor")
372     {
373         // update uniform item's color
374         if ((theColor == WHITE) || (theColor >= GOLD))
375             setObjectColor(swoosh,BLACK);
376         else
377             setObjectColor(swoosh,WHITE);
378
379         // home plate color matches body
380         setObjectColor(uniform.homePlateShow.body,theColor);
381     }
382 }
383
384 if (isZoomed) zoomed.update();
385 }
386
387 function getFont(src)
388 {
389     var theFont = eval("_" + src);
390     eval(src + "Text") = fontDesc[theFont];
391     return fontName[theFont];
392 }
393
394 function setFont(src, theFont)
395 {
396     // update global variable (same as object name)
397     eval("_" + src) = theFont;
398
399     // update description
400     eval(src + "Text") = fontDesc[theFont];
401
402     var frameName = fontName[theFont];
403     if (src.indexOf("num") == 0)
404     {
405         // update uniform number font for front + back on all layers
406         // main, outline, shadow
407         for (props in numFrontImg)
408         {
409             eval("numFrontImg." + props).gotoAndStop(frameName);
410             eval("numBackImg." + props).gotoAndStop(frameName);
411         }
412
413         if (theFont >= INLINE)
414         {
415             _numApp = PRINT;
416             if (_numAccent == SHADOW)
417             {
418                 _numAccent = NONE;
419                 _numAccentColor = NONE;
420             }
421         }
422     }
423 }
```

```
421     }
422   }
423   else if (src.indexOf("name") == 0)
424   {
425     // update each letter in team names top and bottom
426     for (i=1; i<=MAXCHARS; i++)
427     {
428       eval("nameImg.top.char" + i).gotoAndStop(frameName);
429       eval("nameImg.bottom.char" + i).gotoAndStop(frameName);
430     }
431
432     // due to awkward embedding of layers for team name
433     // may need to refresh font, color, and name
434     setName("nameTop",_nameTop);
435     setName("nameBottom",_nameBottom);
436
437     setColor("nameMain", nameMain);
438     setColor("nameAccentColor",_nameAccentColor);
439
440     if (theFont >= INLINE)
441     {
442       _nameApp = PRINT;
443       if (_nameAccent == SHADOW)
444       {
445         _nameAccent = NONE;
446         _nameAccentColor = NONE;
447       }
448     }
449   }
450   else if (src.indexOf("home") == 0)
451   {
452     // home plate font matches
453     uniform.homePlateShow.letter.gotoAndStop(frameName);
454     setChar(uniform.homePlateShow.letter, _homePlateLetter);
455   }
456
457   if (isZoomed) zoomed.update();
458   calculatePrice();
459 }
460
461 function setChar(src, theLetter)
462 {
463   //trace("setChar "+src+"="+theLetter);
464   var upper = theLetter.toUpperCase();
465   var code = upper.charCodeAt(0);
466   if ((code >= CODEA) && (code <= CODEZ))
467   {
468     src.outline.gotoAndStop(code-CODEA+1);
469     src.main.gotoAndStop(code-CODEA+1);
470     src.shadow.gotoAndStop(code-CODEA+1);
471   }
472   else
473   {
474     src.outline.gotoAndStop(CODEZ);
475     src.main.gotoAndStop(CODEZ);
476     src.shadow.gotoAndStop(CODEZ);
477   }
478 }
479
480 function getName(src)
481 {
482   return eval("_"+src);
483 }
484
485 function setName(src, theName)
486 {
487   eval("_" + src) = theName;
488
489   if (src == "saveName") return;
490
491   // src = top or bottom
492   loc = (src == "nameTop") ? "top" : "bottom";
493   imgString = "nameImg." + loc;
494
495   // skewed & rotate place characters depending on odd & even lengths
496   eval(imgString).gotoAndStop((theName.length%2)+1);
497   eval("zoomed."+imgString).gotoAndStop((theName.length%2)+1);
498
499   // pad spaces before to center text
500   for (j=1; j<=Math.floor((MAXCHARS-theName.length))/2; j++)
501   {
502     setChar(eval(imgString + ".char" + (j)), " ");
503     setChar(eval("zoomed."+imgString + ".char" + (j)), " ");
504   }
```



```
505
506 // the actual name string
507 for (i=0; i<theName.length; i++)
508 {
509     setChar(eval(imgString + ".char" + (j+i)), theName.charAt(i));
510     setChar(eval("zoomed."+imgString + ".char" + (j+i)), theName.charAt(i));
511 }
512
513 // pad spaces after to center text
514 for (j+=i; j<=MAXCHARS; j++)
515 {
516     setChar(eval(imgString + ".char" + (j)), " ");
517     setChar(eval("zoomed."+imgString + ".char" + (j)), " ");
518 }
519
520 calculatePrice();
521 }
522
523 function getStyle()
524 {
525     return _nameStyle*FRAMEOFFSET;
526 }
527
528 function setStyle(theStyle)
529 {
530     _nameStyle = theStyle;
531
532     nameStyle.gotoAndStop(theStyle*FRAMEOFFSET);
533     nameImg.gotoAndStop(theStyle*FRAMEOFFSET);
534
535     if (theStyle==BOTTOM)
536     {
537         nameTop.label = "TOP";
538         nameBottom._visible = true;
539     }
540     else
541     {
542         nameTop.label = "NAME";
543         nameBottom._visible = false;
544     }
545
546     setFont("nameFont", _nameFont);
547     setSize("nameSize", _nameSize);
548 }
549
550 function getShow(src)
551 {
552     var toShow = eval("_" + src);
553     eval(src + "Show")._visible = toShow;
554     return toShow;
555 }
556
557 function setShow(src,toShow)
558 {
559     eval("_" + src) = toShow;
560     eval("uniform." + src + "Show")._visible = toShow;
561     if (src == "homePlate")
562     {
563         homePlateSelectedLetter.setEnabled(toShow);
564         homePlateColor.setEnabled(toShow);
565         homePlateFont.setEnabled(toShow);
566
567         zoomed.uniform.homePlateShow._visible = toShow;
568         calculatePrice();
569     }
570 }
571
572 function getApp(src)
573 {
574     return eval("_"+src);
575 }
576
577 function setApp(src,theApp)
578 {
579     eval("_"+src) = theApp;
580     calculatePrice();
581 }
582
583 function getFabric(src)
584 {
585     return _fabric;
586 }
587
588 function setFabric(src,theFabric)
```

```
589 {
590     _fabric = theFabric;
591     calculatePrice();
592 }
593
594 function getAccent(src)
595 {
596     return eval("_" + src);
597 }
598
599 function setAccent(src,theAccent)
600 {
601     eval("_" + src) = theAccent;
602     setColor(src+"Color",eval("_"+src+"Color"));
603     calculatePrice();
604 }
605
606 function getSize(src)
607 {
608     return eval("_" + src);
609 }
610
611 function setSize(src,theSize)
612 {
613     eval("_" + src) = theSize;
614
615     // clear other buttons as radio
616     if (src == "numFrontSize")
617     {
618         zoomed.numFrontImg._xscale = zoomed.numFrontImg._yscale =
619         numFrontImg._xscale = numFrontImg._yscale = theSize/8*100;
620         if (theSize == 4) numFrontSize5.gotoAndStop("off");
621         if (theSize == 5) numFrontSize4.gotoAndStop("off");
622     }
623     else if (src == "numBackSize")
624     {
625         numBackImg._xscale = numBackImg._yscale = theSize/8*100;
626         if (theSize == 6) numBackSize8.gotoAndStop("off");
627         if (theSize == 8) numBackSize6.gotoAndStop("off");
628     }
629     else if (src == "nameSize")
630     {
631         zoomed.nameImg.top._xscale = zoomed.nameImg.top._yscale =
632         nameImg.top._xscale = nameImg.top._yscale = theSize/4*100;
633         zoomed.nameImg.bottom._xscale = zoomed.nameImg.bottom._yscale =
634         nameImg.bottom._xscale = nameImg.bottom._yscale = theSize/4*100;
635         if (theSize == 3) nameSize4.gotoAndStop("off");
636         if (theSize == 4) nameSize3.gotoAndStop("off");
637     }
638
639     calculatePrice();
640 }
641
642 function getLogo()
643 {
644     return _homePlateLetter;
645 }
646
647 function setLogo(theLetter)
648 {
649     _homePlateLetter = theLetter;
650     setChar(uniform.homePlateShow.letter,theLetter);
651     setChar(zoomed.uniform.homePlateShow.letter,theLetter);
652 }
653
654 function getWaistband(src)
655 {
656     return _waistband;
657 }
658
659 function setWaistband(src, theWaistband)
660 {
661     _waistband = theWaistband;
662     calculatePrice();
663 }
664
665 function zoom(isVisible)
666 {
667     isZoomed = isVisible;
668     if (isZoomed)
669     {
670         zoomed.update();
671         zoomed._visible = true;
672         zoomButtonSmall._visible = true;
```

```
673         zoomButtonSmall.gotoAndStop("out");
674         zoomButton.gotoAndStop("out");
675     }
676     else
677     {
678         zoomed._visible = false;
679         zoomButtonSmall._visible = false;
680         zoomButtonSmall.gotoAndStop("in");
681         zoomButton.gotoAndStop("in");
682     }
683 }
684
685 function previewTemplate(uniformStyle)
686 {
687     _uniNum = 0;
688     index = -1;
689     template.gotoAndStop(uniformStyle);
690     template._visible = true;
691     uniform._visible = false;
692
693     if (uniformStyle.indexOf("away")>0)
694         setObjectColor(template.swoosh,WHITE);
695     else
696         setObjectColor(template.swoosh,BLACK);
697 }
698
699 function initTemplate()
700 {
701     uniform.gotoAndStop(template._currentFrame);
702
703     _saveName = "TEAM-NAME";
704     _saveSchool = "";
705     _saveCity = "";
706     _saveState = "";
707
708     isSaved = false;
709
710     _nameTop = "TEAM NAME";
711     _nameBottom = "BOTTOM";
712     _homePlateLetter = "T";
713
714     gotoAndStop("colors");
715     calculatePrice();
716 }
717
718 function initialize()
719 {
720     uniformName = "\"\" + uniformText[_uniformStyle] + "\"\"";
721     uniformNameFull = uniformName + " GAME UNIFORM";
722     hasStockMsg._visible = hasStock(_uniformStyle);
723 }
724
725 function previewUser()
726 {
727     uniform._visible = true;
728
729     uniform.gotoAndStop(uniformFrames[_uniformStyle]+"-user");
730     uniform._visible = true;
731     template._visible = false;
732
733     for (var i=0; i<colorItems.length; i++)
734         setColor(colorItems[i],eval("_" + colorItems[i]));
735
736     setStyle(_nameStyle);
737     setFont("numFont",_numFont);
738     setFont("homePlateFont",_homePlateFont);
739     setLogo(_homePlateLetter);
740
741     setSize("numFrontSize",_numFrontSize);
742     setSize("numBackSize",_numBackSize);
743
744     setShow("homePlate",_homePlate);
745     setShow("rumpLogo",_rumpLogo);
746     setShow("shortLogo",_shortLogo);
747 }
748
749 function initUser(aUniform)
750 {
751     index = aUniform.index;
752
753     _uniNum = aUniform.uniNum;
754     _saveName = aUniform.saveName;
755     _saveSchool = aUniform.saveSchool;
756     _saveCity = aUniform.saveCity;
```

```
757     _saveState = aUniform.saveState;
758
759     isSaved = true;
760     if ((_saveName.length == 0) ||
761         (_saveSchool.length == 0) ||
762         (_saveCity.length == 0) ||
763         (_saveState.length < 2))
764     {
765         isSaved = false;
766     }
767
768     _teamColor1 = aUniform.teamColor1;
769     _teamColor2 = aUniform.teamColor2;
770     _teamColor3 = aUniform.teamColor3;
771     _teamColor4 = aUniform.teamColor4;
772
773     _uniformStyle = aUniform.uniformStyle;
774     _waistband = aUniform.waistband;
775     _fabric = aUniform.fabric;
776     _bodyColor = aUniform.bodyColor;
777     _side1Color = aUniform.side1Color;
778     _side2Color = aUniform.side2Color;
779     _side3Color = aUniform.side3Color;
780     _trim1Color = aUniform.trim1Color;
781     _trim2Color = aUniform.trim2Color;
782     _trim3Color = aUniform.trim3Color;
783     _numFont = aUniform.numFont;
784     _numFrontSize = aUniform.numFrontSize;
785     _numBackSize = aUniform.numBackSize;
786     _numMain = aUniform.numMain;
787     _numAccentColor = aUniform.numAccentColor;
788     _numAccent = aUniform.numAccent;
789     _numApp = aUniform.numApp;
790     _nameStyle = aUniform.nameStyle;
791     _nameTop = aUniform.nameTop;
792     _nameBottom = aUniform.nameBottom;
793     _nameFont = aUniform.nameFont;
794     _nameSize = aUniform.nameSize;
795     _nameMain = aUniform.nameMain;
796     _nameAccentColor = aUniform.nameAccentColor;
797     _nameAccent = aUniform.nameAccent;
798     _nameApp = aUniform.nameApp;
799     _homePlateFont = aUniform.homePlateFont;
800     _homePlateColor = aUniform.homePlateColor;
801     _homePlateLetter = aUniform.homePlateLetter;
802     _homePlate = aUniform.homePlate;
803     _shortLogo = aUniform.shortLogo;
804     _rumpLogo = aUniform.rumpLogo;
805
806     initialize();
807     previewUser();
808 }
809
810 //for (stuff in _root) { if (stuff.charAt(0)=='_') trace(stuff); }
811 //initialize setup based on defaults
812 var _teamColor1, _teamColor2, _teamColor3, _teamColor4;
813 var _uniformStyle, _fabric, _waistband, _bodyColor, _side1Color, _side2Color, _side3Color, _trim1Color, _trim2Color, _trim3Co
    lor;
814 var _numFont, _numFrontSize, _numBackSize, _numMain, _numAccentColor, _numAccent, _numApp;
815 var _nameStyle, _nameTop, _nameBottom, _nameFont, _nameSize, _nameMain, _nameAccentColor, _nameAccent, _nameApp;
816 var _homePlateColor, _homePlateFont, _homePlateLetter, _homePlate, _shortLogo, _rumpLogo;
817 var _uniNum, _saveName;
818
819 gotoAndStop("start");
```

```
1 /*****
2 /* MENS PRICING/CODES */
3 /*****
4
5 isMen = true;
6
7 homePlatePrice = 2.40;
8
9 // garmentPrice [uniform][jersey..shorts + fabric]
10 garmentPrice = [
11     [ 00.00 ],
12     [ 29.00, 33.00, 31.00, 29.00, 36.00, 38.00 ],
13     [ 31.00, 37.00, 32.00, 44.00, 51.00, 52.00, 38.00, 45.00, 47.00 ],
14     [ 30.00, 35.00, 33.00, 43.00, 46.00, 48.00 ],
15     [ 28.00, 34.00, 30.00, 36.00, 41.00, 43.00 ],
16     [ 30.00, 36.00, 33.00, 40.00, 45.00, 47.00 ],
17     [ 27.00, 32.00, 27.00, 30.00, 39.00, 41.00 ],
18     [ 23.00, 29.00, 24.00, 31.00, 37.00, 39.00 ],
19     [ 31.00, 38.00, 32.00, 42.00, 48.00, 50.00 ],
20     [ 30.00, 37.00, 31.00, 29.00, 36.00, 38.00 ],
21     [ 31.00, 36.00, 32.00, 28.00, 35.00, 37.00 ],
22     [ 33.00, 38.00, 33.00, 44.00, 57.00, 59.00 ],
23     [ 25.00, 29.00, 25.00, 39.00, 56.00, 59.00 ]
24 ];
25
26 garmentCode = [
27     [ "(DELETED)" ],
28     [ "NBGM01888", "NBGM01953", "NBGM03607", "NRGM01898", "NRGM01957", "NRGM03615" ],
29     [ "NBGM01889", "NBGM01954", "NBGM03610", "NRGM01899", "NRGM01958", "NRGM03618", "NRGM02023", "NRGM02024", "NRGM04290" ],
30     [ "NBGM01891", "NBGM01956", "NBGM03606", "NRGM01902", "NRGM01960", "NRGM03614" ],
31     [ "NBGM01895", "NBGM01892", "NBGM03608", "NRGM01906", "NRGM01903", "NRGM03616" ],
32     [ "NBGM01894", "NBGM01893", "NBGM03609", "NRGM01905", "NRGM01904", "NRGM03617" ],
33     [ "NBGM06642", "NBGM06644", "NBGM06643", "NRGM06645", "NRGM06647", "NRGM06646" ],
34     [ "NBGM02094", "NBGM01897", "NBGM03611", "NRGM02115", "NRGM02116", "NRGM03619" ],
35     [ "NBGM03690", "NBGM03688", "NBGM03689", "NRGM03696", "NRGM03694", "NRGM03695" ],
36     [ "NBGM03693", "NBGM03691", "NBGM03692", "NRGM03699", "NRGM03697", "NRGM03698" ],
37     [ "NBGM06654", "NBGM06656", "NBGM06655", "NRGM06657", "NRGM06659", "NRGM06658" ],
38     [ "NBGM06636", "NBGM06638", "NBGM06637", "NRGM06639", "NRGM06641", "NRGM06640" ],
39     [ "NBGM06648", "NBGM06650", "NBGM06649", "NRGM06651", "NRGM06653", "NRGM06652" ]
40 ];
41
42 // [font][solid..accent]
43 numPrintPrice = [[ 2.30, 4.30 ],[ 2.40, 4.90 ]];
44 namePrintPrice = [ 2.30, 4.30 ];
45
46 // [size][solid,outline,shadow]
47 numTwillPrice = [
48     [ ],[ ],[ ],[ ],
49     [ 1.40, 2.40, 3.00 ],
50     [ 1.50, 2.80, 3.20 ],
51     [ 1.60, 3.00, 3.50 ],
52     [ ],
53     [ 2.10, 3.90, 5.20 ]
54 ];
55
56 nameTwillPrice = [
57     [ ],[ ],[ ],
58     [ 1.40, 2.50, 2.60 ],
59     [ 1.50, 2.80, 3.00 ]
60 ];
61
62 uniformText = [
63     "(DELETED)", "RED STORM", "CARDINAL", "HARTFORD", "TUCSON",
64     "COWBOY", "OWL", "(DELETED)", "BLUE DEVIL", "LEXINGTON",
65     "INDY", "CHAMPAIGN", "JAYHAWK", "LONGHORN"
66 ];
67
68 uniformFrames = [
69     "(Cinci)", "John", "SU", "UConn", "AZ",
70     "OK", "Temple", "American", "Duke", "KY",
71     "IN", "Ill", "KS", "TX"
72 ];
73
74 // uniform constants
75 STJOHN = 1;
76 STANFORD = 2;
77 UCONN = 3;
78 ARIZONA = 4;
79 OKLAHOMA = 5;
80 DUKE = 8;
81 KENTUCKY = 9;
82
83 CINCINNATI = 0; // 01/03 deleted
```

```
84 AMERICAN = 7;          // 01/03 deleted
85 TEMPLE = 6;            // 01/03 updated
86 INDIANA = 10;          // 01/03 added
87 ILLINOIS = 11;         // 01/03 added
88 KANSAS = 12;           // 01/03 added
89 TEXAS = 13;            // 01/03 added
90
91 function hasWaistband()
92 {
93     if ((_uniformStyle == CINCINNATI) || (_uniformStyle == STANFORD))
94         return true;
95     else
96         return false;
97 }
98
99 function hasStock()
100 {
101     if ((_uniformStyle == ARIZONA) || (_uniformStyle == STANFORD))
102         return true;
103     else
104         return false;
105 }
106
107 function getEnabled(target)
108 {
109     if ((target._name.indexOf("home") == 0) && (target != homePlateAvailable))
110         target.setEnabled(_homePlate);
111     else
112         target.setEnabled(true);
113 }
114
115 function setEnabled(target, isEnabled)
116 {
117     target.gotoAndStop(isEnabled?"on":"off");
118     target.setOptions(NULL);
119 }
120
121
```

login

```
1 on (release, keyPress "<Enter>")
2 {
3     user = _root.user;
4     password = _root.password;
5
6     if ((user == "") || (password == ""))
7     {
8         _root.msg = "User and password may not be empty.";
9     }
10    else
11    {
12    //        getUrl("login.asp", "_blank", "GET");
13        _root.msg = "Logging in ...";
14        loadVariables ("login.asp", _root.loginButton, "POST");
15    }
16 }
```

```
1 onClipEvent(data)
2 {
3     CusNum = parseInt(CusNum);
4     if (CusNum >= 0)
5     {
6         _root.msg = "Login successful.";
7         _root._cusNum = CusNum;
8
9         thexml = new XML();
10        thexml.onLoad = _root.getSavedList;
11        thexml.load("load.asp");
12        _root.start = getTimer();
13
14        _root.play();
15    }
16    else
17    {
18        _root.msg = "Login unsuccessful.";
19    }
20 }
```

textfield

```
1 onClipEvent (load)
2 {
3     current = _root.getName(_parent._name);
4     _parent.text = current;
5 }
6
7 onClipEvent (enterFrame)
8 {
9     if (current == _parent.text)
10        return;
11    _parent.text = _parent.text.toUpperCase();
12    for (var i=0; i<_parent.text.length; i++)
13    {
14        if ((_parent.text.charCodeAt(i) < _root.CODEA) || (_parent.text.charCodeAt(i) > _root.CODEZ))
15        {
16            _parent.text = _parent.text.substr(0,i) + " " + _parent.text.substr(i+1);
17        }
18    }
19    current = _parent.text;
20    _root.setName(_parent._name,current);
21 }
```

save

```
1 on (release)
2 {
3     // prevents saving multiple time while connecting to server
4     if (!_root.saveMsg.done)
5         return;
6
7     saveName = _root._saveName;
8     saveSchool = _root._saveSchool;
9     saveCity = _root._saveCity;
10    saveState = _root._saveState;
11
12    if ((saveName.length == 0) ||
13        (saveSchool.length == 0) ||
14        (saveCity.length == 0) ||
15        (saveState.length < 2))
16    {
17        _root.saveMsg.gotoAndPlay("validate");
18        _root.isSaved = false;
19        _root.reviewButton.setEnabled(false);
20        return;
21    }
22
23    _root.saveMsg.done = false;
24    _root.saveMsg.gotoAndPlay("saving");
```

```

25
26     CusNum = _root._cusNum;
27     UniNum = 0;
28
29     if ((_root.index>=0) && (saveName == _root.savedUniforms[_root.index].saveName))
30         UniNum = _root.savedUniforms[_root.index].UniNum;
31
32     if (UniNum == 0)
33         _root.index = -1;
34
35     teamColor1 = _root._teamColor1;
36     teamColor2 = _root._teamColor2;
37     teamColor3 = _root._teamColor3;
38     teamColor4 = _root._teamColor4;
39
40     uniformStyle = _root._uniformStyle;
41     waistband = _root._waistband;
42     fabric = _root._fabric;
43     bodyColor = _root._bodyColor;
44     side1Color = _root._side1Color;
45     side2Color = _root._side2Color;
46     side3Color = _root._side3Color;
47     trim1Color = _root._trim1Color;
48     trim2Color = _root._trim2Color;
49     trim3Color = _root._trim3Color;
50     numFont = _root._numFont;
51     numFrontSize = _root._numFrontSize;
52     numBackSize = _root._numBackSize;
53     numMain = _root._numMain;
54     numAccentColor = _root._numAccentColor;
55     numAccent = _root._numAccent;
56     numApp = _root._numApp;
57     nameStyle = _root._nameStyle;
58     nameTop = _root._nameTop;
59     nameBottom = _root._nameBottom;
60     nameFont = _root._nameFont;
61     nameSize = _root._nameSize;
62     nameMain = _root._nameMain;
63     nameAccentColor = _root._nameAccentColor;
64     nameAccent = _root._nameAccent;
65     nameApp = _root._nameApp;
66     homePlateFont = _root._homePlateFont;
67     homePlateColor = _root._homePlateColor;
68     homePlateLetter = _root._homePlateLetter;
69     homePlate = _root._homePlate;
70     shortLogo = _root._shortLogo;
71     rumpLogo = _root._rumpLogo;
72
73 //     getURL("save.asp", "_blank", "GET");
74     loadVariables("save.asp", _root.saveButton, "POST");
75 }

```

```

1  onClipEvent(data)
2  {
3      saved = (saved.indexOf("true")>=0);
4      UniNum = parseInt(UniNum);
5
6      _root.saveMsg.saved = saved?"true":"false";
7      _root.saveMsg.done = true;
8      _root.isSaved = true;
9      _root.reviewButton.setEnabled(true);
10
11     if (saved)
12     {
13         theUniform = new Array();
14         for (var prop in this)
15         {
16             eval("theUniform."+prop) = eval(prop);
17         }
18
19         if (_root.index < 0)
20         {
21             _root.index = theUniform.index = _root.savedUniforms.length;
22             _root._uniNum = theUniform.UniNum = UniNum;
23             _root.savedUniforms.push(theUniform);
24         }
25         else
26         {
27             theUniform.index = _root.index;
28             _root.savedUniforms[_root.index] = theUniform;
29         }
30     }
31 }

```



```

print
1 stop();
2 if (!uniform._visible) previewUser();
3
4 function catalogFontName(font, accent)
5 {
6     theFontName = fontNames[font];
7     if (font == DOUBLE)
8         theFontName = "FULL BLOCK OUTLINE";
9     else if (accent == SHADOW)
10    {
11        if (font == GOTHIC)
12            theFontName = "ATHLETIC GOTHIC SHADOW";
13        else if (font == NIKEBLOCK)
14            theFontName = "ATHLETIC SHADOW";
15        else if (font == FULL)
16            theFontName = "FULL BLOCK SHADOW";
17    }
18
19    theFontName += " ";
20    theFontName += (accent == NONE) ? "1C" : "2C";
21    return theFontName;
22 }
23
24 function getSpecs()
25 {
26 // uniform specs
27 //styleSpec = "Style / "+uniformText[_uniform];
28 jerseySpec = "Jersey Style / #"+garmentCode[_uniformStyle][_fabric];
29 shortsSpec = "Shorts Style / #"+garmentCode[_uniformStyle][_fabric+3*(_waistband+1)];
30 fabricSpec = "Fabric / "+fabricStyles[_fabric];
31 bodySpec = "Color / "+colorText[_bodyColor];
32 insertSpec = "Inserts / "+colorText[_side1Color]+
33     (_side2Color?"-"+colorText[_side2Color]: "");
34 trimSpec = "Trims / "+colorText[_trim1Color]+
35     (_trim2Color?"-"+colorText[_trim2Color]: "");
36
37 // number specs
38 numFontSpec = "Font / "+catalogFontName(_numFont, _numAccent);
39 numSizeSpec = "Size / Front-"+_numFrontSize+"\\" Back-"+_numBackSize+"\\";
40 numAppSpec = "Application / "+((_numApp==PRINT)? "PRINT": "TACKLE TWILL");
41 numColorSpec = "Colors / "+colorText[_numMain]+
42     (_numAccentColor?"-"+colorText[_numAccentColor]: "");
43
44 // team name specs
45 nameSpec = "Name / "+_nameTop+
46     (_nameStyle==BOTTOM?" "+_nameBottom: "");
47 nameStyleSpec = "Style / " + nameStyles[_nameStyle];
48 nameFontSpec = "Font / "+catalogFontName(_nameFont, _nameAccent);
49 nameSizeSpec = "Size / "+_nameSize+"\\";
50 nameAppSpec = "Application / "+((_nameApp==PRINT)? "PRINT": "TACKLE TWILL");
51 nameColorSpec = "Colors / "+colorText[_nameMain]+
52     (_nameAccentColor?"-"+colorText[_nameAccentColor]: "");
53
54 // custom logo specs
55 homeSpec = "Homeplate / "+(_homePlate?(colorText[_homePlateColor]+"-"+_homePlateLetter): "NO");
56 homeFontSpec = "Font / "+(_homePlate?fontNames[_homePlateFont]+" SOLID": "NONE");
57 logoSpec = "Logos / Shorts-"+(_shortLogo?"YES": "NO")+" Rump-"+(_rumpLogo?"YES": "NO");
58 }
59
60 getSpecs();

```

```

color palette
1 stop();
2
3 function setEnabled(isEnabled)
4 {
5     gotoAndStop(isEnabled?"on":"off");
6
7     buttonArray = new Array(color1, color2, color3, color4);
8     frameArray = new Array(frame1, frame2, frame3, frame4);
9     theColor = _root.getColor(_name);
10    selected = null;
11
12    for (i=0; i<buttonArray.length; i++)
13    {
14        button = buttonArray[i];
15        button.theColor = _root.teamColors[i];
16        _root.setObjectColor(button,_root.teamColors[i]);
17
18        if (_root.teamColors[i] == _root.NONE)
19        {
20            button._visible = false;
21            frameArray[i]._visible = false;
22        }
23        else if (theColor == _root.teamColors[i])
24        {
25            button._visible = true;
26            frameArray[i]._visible = true;
27            selected = button;
28        }
29    }
30
31    if (isEnabled && selected != null)
32    {
33        hilite._visible = true;
34        hilite._x = selected._x;
35        hilite._y = selected._y;
36        text = _root.colorText[theColor];
37    }
38    else
39    {
40        hilite._visible = false;
41    }
42 }
43
44 _root.setEnabled(this);

```