

Main.as

```
1 package
2 {
3     import com.vaden.ui.PlayerBase;
4
5     import flash.display.Sprite;
6
7     [SWF(width="760",height="500",frameRate="30", backgroundColor="0x000000")]
8
9     public class Main extends Sprite
10    {
11        public function Main()
12        {
13            var player:PlayerBase = new PlayerBase();
14            player.x = (stage.stageWidth/2) - (player.width/2);
15            player.y = (stage.stageHeight/2) - (player.height/2);
16            this.addChild(player);
17        }
18    }
19 }
```

Bars.as

```
1 package com.vaden.ui
2 {
3     import libs.Bars;
4
5     public class Bars extends libs.Bars
6     {
7         public function Bars()
8         {
9             super();
10
11             this.mask = this.bars_mask;
12         }
13     }
14 }
```

Controls.as

```
1 package com.vaden.ui
2 {
3     import com.vaden.managers.RollManager;
4
5     import libs.Controls;
6
7     public class Controls extends libs.Controls
8     {
9
10        public function Controls()
11        {
12            super();
13            this.btn_play.stop();
14            this.btn_mute.stop();
15            this.btn_next.stop();
16
17        }
18    }
19 }
20 }
```

DisplayPanel.as

```
1 package com.vaden.ui
2 {
3     import com.vaden.events.ImageEvent;
4     import com.vaden.loaders.ImageLoaderBasic;
5
6     import flash.text.TextField;
7
8     import libs.DisplayPanel;
9
10    public class DisplayPanel extends libs.DisplayPanel
11    {
12        private var _imageList:Array;
13        private var _currentImage:int;
14        private var _ld:ImageLoaderBasic;
15        private var _titleText:TextField;
16        private var _artistText:TextField;
17        private var _path:String;
18
19        public function DisplayPanel()
20        {
21            init();
22        }
23
24        private function init():void
25        {
26            _imageList = [ ];
27            _currentImage = 0;
28        }
29
30        private function loadImg():void
31        {
32            _ld = new ImageLoaderBasic(_path + _imageList[_currentImage].art);
33            _ld.addEventListener(ImageEvent.IMAGE_LOADED, onLoad);
34            _titleText.text = _imageList[_currentImage].title + "";
35            _artistText.text = _imageList[_currentImage].artist + "";
36
37        }
38
39        private function onLoad(e:ImageEvent):void
40        {
41            if(this.numChildren > 0)
42            {
43                this.removeChildAt(0);
44            }
45            this.addChild(e.image);
```

DisplayPanel.as

```
46     }
47
48     public function display():void
49     {
50         loadImg();
51     }
52
53     public function next():void
54     {
55         if(this.numChildren > 0)
56         {
57             this.removeChildAt(0);
58         }
59         _currentImage++;
60         if(_currentImage == _imageList.length)
61         {
62             _currentImage = 0;
63         }
64         _ld = new ImageLoaderBasic(_path + _imageList[_currentImage].art);
65         _ld.addEventListener(ImageEvent.IMAGE_LOADED, onLoad);
66         _titleText.text = _imageList[_currentImage].title + "";
67         _artistText.text = _imageList[_currentImage].artist + "";
68     }
69
70     public function previous():void
71     {
72         _currentImage--;
73         if(_currentImage < 0)
74         {
75             _currentImage = _imageList.length - 1;
76         }
77         _ld = new ImageLoaderBasic(_path + _imageList[_currentImage].art);
78         _ld.addEventListener(ImageEvent.IMAGE_LOADED, onLoad);
79         _titleText.text = _imageList[_currentImage].title + "";
80         _artistText.text = _imageList[_currentImage].artist + "";
81     }
82
83     public function set path(value:String):void
84     {
85         _path = value;
86     }
87
88     public function set imageList(value:Array):void
89     {
90         _imageList = value;
```

DisplayPanel.as

```
91     }
92
93     public function set title(value:TextField):void
94     {
95         _titleText = value;
96     }
97
98     public function set artist(value:TextField):void
99     {
100         _artistText = value;
101     }
102 }
103 }
104
105
106
107
108
109
110
111
112
113
114
115
116
117
```

InfoPanel.as

```
1 package com.vaden.ui
2 {
3     import libs.InfoPanel;
4
5     public class InfoPanel extends libs.InfoPanel
6     {
7         public function InfoPanel()
8         {
9             super();
10
11             this.tfArtist.text = "";
12             this.tfTitle.text = "";
13             this.tfTime.text = "";
14         }
15     }
16 }
```

PlayerBase.as

```
1 package com.vaden.ui
2 {
3     import com.vaden.managers.SliderManager;
4     import com.vaden.utils.ConversionUtils;
5     import com.vaden.vo.MediaVO;
6
7     import flash.display.Sprite;
8     import flash.events.Event;
9     import flash.events.MouseEvent;
10    import flash.media.Sound;
11    import flash.media.SoundChannel;
12    import flash.media.SoundTransform;
13    import flash.net.URLLoader;
14    import flash.net.URLRequest;
15
16    import libs.PlayerBase;
17
18    public class PlayerBase extends libs.PlayerBase
19    {
20        private var _controls:Controls;
21        private var _info:InfoPanel;
22        private var _music:Sound;
23        private var _sch:SoundChannel;
24        private var _st:SoundTransform;
25        private var _sndpos:int;
26        private var _playerData:Array;
27        private var _vo:MediaVO;
28        private var _currentSong:int;
29        private var _path:String;
30        private var _display:DisplayPanel;
31        private var _musicPlay:Boolean;
32        private var _sm:SliderManager;
33        private var _bars:Bars;
34
35        public function PlayerBase()
36        {
37            init();
38            loadXML();
39        }
40
41        private function init():void
42        {
43            _controls = new Controls();
44            _controls.x = 200;
45            _controls.y = 55.00;
```

PlayerBase.as

```
46     this.addChild(_controls);
47
48     _controls.btn_next.addEventListener(MouseEvent.CLICK, onNext);
49     _controls.btn_prev.addEventListener(MouseEvent.CLICK, onPrev);
50     _controls.btn_mute.addEventListener(MouseEvent.CLICK, onMute);
51
52
53     var slider:Slider = new Slider();
54     slider.x = 200;
55     slider.y = 115.20;
56     this.addChild(slider);
57
58
59     _sm = new SliderManager();
60     _sm.setUpAssets(slider.mc_track, slider.mc_handle);
61     _sm.percent = .5;
62     _sm.addEventListener(Event.CHANGE, onComplete);
63
64     _path = "assets/audio/";
65     _currentSong = 0;
66     _musicPlay = false;
67     _controls.btn_play.addEventListener(MouseEvent.MOUSE_DOWN, onPlay);
68
69     _st = new SoundTransform();
70     _st.volume = .5;
71
72     _bars = new Bars();
73     _bars.x = 402.75;
74     _bars.y = 14.45;
75     this.addChild(_bars);
76 }
77
78 private function loadXML():void
79 {
80     //load the XML data from the XML file and parse the data...
81     var urlLoader:URLLoader = new URLLoader();
82     urlLoader.load(new URLRequest("assets/xml/playlist.xml"));
83     urlLoader.addEventListener(Event.COMPLETE, parseData);
84 }
85
86 private function parseData(e:Event):void
87 {
88     //creates the _playerData array to hold the data from the XML file
89     _playerData = [ ];
90     var xmlData:XML = XML(e.target.data);
```

PlayerBase.as

```
91      //takes all of the items in the xml data and assigns the
92      //corresponding VO to them and pushes them into the array
93      for each (var data:XML in xmlData.SONG)
94      {
95          _vo = new MediaVO();
96          _vo.art = data.@ART;
97          _vo.artist = data.@ARTIST;
98          _vo.mp3 = data.@MP3;
99          _vo.title = data.@TITLE;
100         _playerData.push(_vo);
101     }
102     _display = new DisplayPanel();
103     _display.x = 12.50;
104     _display.y = 14.00;
105     _display.path = "assets/images/";
106     _display.imagelist = _playerData;
107     this.addChild(_display);
108
109     _info = new InfoPanel();
110     _info.x = 14;
111     _info.y = 100;
112     this.addChild(_info);
113
114     _display.title = _info.tfTitle;
115     _display.artist = _info.tfArtist;
116     _display.display();
117
118
119
120     // _info.tfTime.text =
121 }
122
123 private function onNext(e:MouseEvent):void
124 {
125     if (_musicPlay == true)
126     {
127         _sch.stop();
128         _display.next();
129         _currentSong++;
130         if (_currentSong == _playerData.length)
131         {
132             _currentSong = 0;
133         }
134         _music = new Sound();
```

PlayerBase.as

```
135         _music.load(new URLRequest(_path + _playerData
[_currentSong].mp3 ));
136         _sch = _music.play();
137     }
138     else
139     {
140         _display.next();
141         _currentSong++;
142     }
143 }
144
145 private function onPrev(e:MouseEvent):void
146 {
147     if (_musicPlay == true)
148     {
149         _sch.stop();
150         _display.previous();
151         _currentSong--;
152         if (_currentSong < 0 )
153         {
154             _currentSong = _playerData.length - 1;
155         }
156         _music = new Sound();
157         _music.load(new URLRequest(_path + _playerData
[_currentSong].mp3 ));
158         _sch = _music.play();
159     }
160     else
161     {
162         _display.previous();
163         _currentSong--;
164     }
165 }
166
167 private function onPlay(e:MouseEvent):void
168 {
169     if (_controls.btn_play.currentFrame == 1)
170     {
171         _controls.btn_play.gotoAndStop(2);
172         _music = new Sound();
173         _music.load(new URLRequest(_path + _playerData
[_currentSong].mp3 ));
174         _sch = _music.play(_sndpos,0,_st);
175         _musicPlay = true;
176         _sch.addEventListener(Event.SOUND_COMPLETE, onDone);
```

PlayerBase.as

```
177         this.addEventListener(Event.ENTER_FRAME, update);
178     }
179     else
180     {
181         _sndpos = _sch.position;
182         _controls.btn_play.gotoAndStop(1);
183         _sch.stop();
184         _musicPlay = false;
185     }
186 }
187
188 private function update(e:Event):void
189 {
190     if (_sch)
191     {
192         var time:String = ConversionUtils.toTimeCode(_sch.position);
193         _info.tfTime.text = time;
194         trace(_sch.leftPeak);
195
196         _bars.coverLeft.y = _sch.leftPeak * -131;
197         _bars.coverRight.y = _sch.rightPeak * -131;
198     }
199 }
200
201 private function onMute(e:MouseEvent):void
202 {
203     if(_controls.btn_mute.currentFrame == 1)
204     {
205         _controls.btn_mute.gotoAndStop(2);
206         _st = new SoundTransform();
207         _st.volume = 0;
208         _sch.soundTransform = _st;
209     }
210     else
211     {
212         _controls.btn_mute.gotoAndStop(1);
213         _st.volume = _sm.percent;
214         _sch.soundTransform = _st;
215     }
216 }
217
218 private function onDone(e:Event):void
219 {
220     _musicPlay = true;
221     _currentSong++;
```

PlayerBase.as

```
222         _sch.removeEventListener(Event.SOUND_COMPLETE, onDone);
223         _display.next();
224         _music = new Sound();
225         _music.load(new URLRequest(_path + _playerData[_currentSong].mp3 ));
226         _sch = _music.play();
227         _sch.addEventListener(Event.SOUND_COMPLETE, onDone);
228     }
229
230     private function onComplete(e:Event):void
231     {
232         _st = new SoundTransform();
233         _st.volume = _sm.percent;
234         _sch.soundTransform = _st;
235     }
236 }
237 }
```

Slider.as

```
1 package com.vaden.ui
2 {
3
4     import libs.Slider;
5
6     public class Slider extends libs.Slider
7     {
8
9         public function Slider()
10        {
11            super();
12        }
13    }
14 }
```