TWaver HTML5 Training

servasoftware

Training Objectives

- TWaver and HTML5 Background
- TWaver Design Patterns
- TWaver Components
- Beyond TWaver

Training Outline

- Introduce TWaver and HTML5
- Recommendation for learning
- Step-by-Step examples
- TWaver core components
- Knowledge you need beyond TWaver
- Q&A

- Why HTML5?
- Why TWaver?
- Is TWaver HTML5 ready?

Why HTML5?

HT

- Fast-growing, for the future
- Cross-platform PC, Mac, Mobile ...
- Light weight, dynamic, node.js ...

Why TWaver?



- Product Family
- Design Pattern
- Easy, no need to write low-level code

Product Family

- Java
- Web
- Flex
- .NET
- HTML5...



TWaver Infrastructure



Performance

- twaver.network.Network Less than 2,000 nodes or links are suggested
- twaver.canvas.Network Less than 10,000 nodes or links are suggested



TWaver HTML5 Performance Report

Ready

- Java -> Web(SVG) -> Flex -> HTML5
- Demos, Develop guide, API docs, Forum ...
- Chrome, Firefox, Safari, IE9, Opera
- iOS, android, windows phone 7, webOS ...

How to Learn

- HTML, CSS, Javascript, <canvas>
- HTML5, CSS3 ...
- Firefox, IE, Safari, Chrome, mobile ...

Hello Canvas



HTML + CSS + Canvas + Javascript

IDE

/**		
* @constructor		
*/		
<pre>twaver.Node = function(){}; /**</pre>		
* gets the node's height.		
* @return {Number}. */		
<pre>twaver.Node.prototype.getHeight = function /**</pre>	O {}	
* gets the node's width.		
<pre>* @return {Number}. */</pre>		
<pre>twaver.Node.prototype.getWidth = function (/**</pre>	(} ()	•
* gets the node's parent node.		
<pre>* @return {twaver.Data}. */</pre>	_	
<pre>twaver.Node.prototype.getParent = function</pre>	var node	= ne
	node.get	-
		J _o get
		Jo get
		Jo get

- Plain Text Editor
- Spket, Netbeans, VS2010...
- jsdoc

twaver.Node.prototype.getParent = function	<pre>var node = new twaver.Node(); node.get</pre>	
	f_0 getHeight	Number getHeight()
JS Doc	f_{0} getImage f_{0} getLinks f_{0} getParent f_{0} getWidth	gets the node's height. Returns: {Number}.

Browsers and Debug

- Webkit -> Web Inspector
- Firefox -> firebug
- IE9 -> developer tools
- Opera -> dragonfly
- console.log(...);

-	o 🗠			~		30		_			
Elem	ients Resources	Network	Scripts	Timeline					Searc	h Scripts	
4	► 01%20-%2	Ohello_can	vas.html	\$		н	2	÷	1	n/a	
57)				ń	W W	atch	Expre	essio	ns	
58	function dr	awRandom	Star(){			_		-	-		
653	g.savet	/; late(Mat	h, candor	0 + 0	620			(A	dd) (Refresh)
61	var sca	le = Mat	h. randor	0 * M	th.	YC	all Sta	ack.			
62	g.scale	(scale,	scale);			10	an su	IL.N.			
63	drawSta	r(g);							Not I	laused	
64	g.fillS	tyle = "	white";			¥ 50	ope \	Varia	bles		
60	g.shado	wColor =	-red-;				cope.				
67	g.shado	wotur =	10;						Not I	aused	
68	g.resto	re():			1	V Br	reakp	oints			
69	}					M 01	\$20-3	20hel	lo car	vas.html-60	0
78						- a	tran	slat	e (Ma	th, rando	
71	function dr	awStar(g	14			- W	orker				ODah
72	var gap	Path()				P 11	UTKEI	2			Uneon
74	g. soyeT	o(8, -1)	-		- U						
75	g.lineT	olgap	(qeo								
76	g.lineT	o(1, 0);			Ŧ	1					
77 7						1					

How to Learn

- «Object-Oriented JavaScript»
- «Pro JavaScript Design Patterns»
- http://www.w3.org/TR/CSS2/
- http://www.w3.org/TR/html5/
- http://html5weekly.com/
- http://javascriptweekly.com/
- http://twaver.servasoft.com/





more references

- <u>http://html5rocks.com</u>
- http://www.whatwg.org/
- https://developer.mozilla.org/en/
- <u>http://developer.apple.com/library/safari</u>
- <u>http://jQuery.com</u>

Step by Step

- DataBox, ElementBox
- Data, Node, Link, Group ...
- Network, Tree, Table, Sheet, Charts
- renderer and editor, listeners, interaction
- TabPane, SplitPane, CSS
- Alarms
- Beyond TWaver

Package Contents

🔻 🛅 twaver-html5-beta		IS Doc
twaver.sdoc	619 KB	Je Bee
🕨 🚞 alarm		
🕨 🚞 chart		
editor		
images		Demos
network		
table		
tree		
demo.html	3 KB	
readme.html	264 bytes	
Js demo.js	78 KB	twaveris
twaver.js	630 KB	

Step I - box, node, link, network

```
<!DOCTYPE html>
<html>
<head>
                <title>TWaver HTML5 Step by Step</title>
                <script type="text/javascript" src="twaver.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></scr
                <<u>script</u> type="text/javascript">
                                function init() {
                                                var box = new twaver.ElementBox();
                                                var node1 = new twaver.Node();
                                                var node2 = new twaver.Node();
                                                var link = new twaver.Link(node1, node2);
                                                node1.setLocation(50, 50);
                                                node2.setLocation(200, 150);
                                                box.add(node1);
                                                box.add(node2);
                                                box.add(link);
                                                var network = new twaver.network.Network(box);
                                                document.body.appendChild(network.getView());
                                                network.getView().style.background = '#E9E9E9';
                                                network.adjustBounds({x: 10, y: 10, width: 300, height: 300});
                </script>
</head>
<body onload="init()">
</body>
</html>
```

Step2 - tree, table

var column = new twaver.Column(); column.setWidth(260); column.setPropertyType('field'); column.setPropertyName('_id'); table.getColumnBox().add(column);

```
column = new twaver.Column();
column.setPropertyType('accessor');
column.setPropertyName('name');
table.getColumnBox().add(column);
```

```
column = new twaver.Column();
column.setPropertyType('style');
column.setValueType('color');
column.setPropertyName('select.color');
table.getColumnBox().add(column);
```

column = new twaver.Column(); column.setPropertyType('client'); column.setPropertyName('producer'); table.getColumnBox().add(column);

Step2 - tree, table



Tree



Table

ID	Logo	Color	Web Site
SERVA Gro	SERVER		www.servagroup.com
USA			
WPT Power	(711)		www.wptpower.com
China	-		
SERVA SJS	515		www.sjs.servacorp.com
SERVA Ser	4.		www.servasoftware.com
TWaver	THaver'		www.servasoftware.c
Waver Java	TH		www.servasoftware.c
TWaver Web	TH		www.servasoftware.c
Waver Flex	TH		www.servasoftware.c
TMOUGE NET			www.convocoftworo.c

Name	Inner Color	-
TWaver-76-6		
TWaver-42-2		
TWaver-30-0		
TWaver-19-9		*
TWaver-4-4		
TWaver-22-2		
THIS SO O	12 C	1000

Name	Inner Color 🔺
TWaver-14-4	
TWaver-43-3	
TWaver-37-7	
TWaver-67-7	
TWaver-26-6	
TWaver-80-0	

Step3 - tree table, group, selection color

```
var node1 = new twaver.Node();
var node2 = new twaver.Node();
var link = new twaver.Link(node1,
node2);
var group = new twaver.Group();
```

```
// parent & children
group.addChild(node1);
node2.setParent(group);
```

```
// accessor property
node1.setName('from');
node2.setName('to');
link.setName('link');
group.setName('group');
node1.setLocation(50, 50);
node2.setLocation(200, 150);
```

```
// style property
node1.setStyle('select.color',
'yellow');
node2.setStyle('select.color', 'red');
link.setStyle('select.color', 'blue');
```

```
// change Table -> TreeTable
var table = new
twaver.controls.TreeTable(box);
```

. . .

Step3 - tree table, group, selection color



Step4 - sheet

```
var sheet = new
twaver.controls.PropertySheet(box);
document.body.appendChild(sheet.getVie
w());
sheet.getView().style.outline = '1px
solid #E9E9E9':
sheet.adjustBounds({
            x : 320,
            y : 160,
            width : 200,
            height : 150
        });
sheet.setEditable(true);
var property = new twaver.Property();
property.setName('Name');
property.setPropertyType('accessor');
property.setPropertyName('name');
property.setEditable(true);
sheet.getPropertyBox().add(property);
property = new twaver.Property();
```

property.setName('Color');

```
property.setPropertyType('style');
property.setValueType('color');
property.setPropertyName('select.color
');
property.setEditable(true);
sheet.getPropertyBox().add(property);
property = new twaver.Property();
property.setName('Location');
property.setPropertyType('accessor');
property.setPropertyName('location');
property.setEditable(true);
property.getValue = function(data) {
    return 'x:' +
parseInt(data.getX()) + ', y:' +
parseInt(data.getY());
}:
property.isVisible = function(data) {
    return data instanceof
twaver.Node;
}:
sheet.getPropertyBox().add(property);
```

Step4 - sheet



Step4 - table header

```
// wrap table with TablePane
var table = new twaver.controls.TreeTable(box);
var tablePane = new twaver.controls.TablePane(table);
tablePane.getTableHeader().getView().style.background = 'url(th.gif) repeat-x';
document.body.appendChild(tablePane.getView());
tablePane.getView().style.background = '#E9E9E9';
tablePane.adjustBounds({
            x : 10,
            y : 320,
            width : 510,
            height : 160
        });
var column = table.getTreeColumn();
column.setName('Hierarchy');
column.renderHeader = function(div) {
    var href = document.createElement('a');
    href.setAttribute('href', 'http://twaver.servasoft.com/forum/');
    href.setAttribute('target', '_blank');
    href.innerHTML = 'TWaver Forum';
    div.style.textAlign = 'center';
    div.appendChild(href);
};
```

Step4 - table header



Step5 - List and listeners

```
var list = new twaver.controls.List();
document.body.appendChild(list.getView());
list.getView().style.background = '#E9E9E9';
list.adjustBounds({
      x : 530,
     v : 10,
     width : 150.
      height : 470
    });
network.addInteractionListener(function(e) {
      var data = new twaver.Data();
      data.setIcon(null);
      data.setName(e.kind);
      var listBox = list.getDataBox();
      listBox.add(data);
listBox.getSelectionModel().setSelection(data)
,
      if (listBox.size() > 24) {
listBox.remove(listBox.getDatas().get(0));
    });
list = new twaver.controls.List();
document.body.appendChild(list.getView());
```

```
list.getView().style.background = '#E9E9E9';
list.adjustBounds({
      x : 690,
      y : 10,
      width : 170,
      height: 470
    });
box.addDataPropertyChangeListener(function(e)
{
      if (e.property === 'location') {
        var name = e.source.getName() + '
location x:'
            + parseInt(e.newValue.x) + ' y:'
            + parseInt(e.newValue.y);
        var data = new twaver.Data();
        data.setIcon(null);
        data.setName(name);
        var listBox = list.getDataBox();
        listBox.add(data);
listBox.getSelectionModel().setSelection(data)
        if (listBox.size() > 24) {
listBox.remove(listBox.getDatas().get(0));
```

Step5 - List and listeners



Step6 - charts - PieChart

```
function initPieChart(box) {
    var xData = new twaver.Element();
    var yData = new twaver.Element();
    var chart = new twaver.charts.PieChart();
    chart.getDataBox().add(xData);
    chart.getDataBox().add(yData);
```

```
document.body.appendChild(chart.getView());
chart.getView().style.background = '#E9E9E9';
chart.adjustBounds({
```

```
x : 690,
y : 10,
width : 170,
height : 150
});
```

}



```
box.addDataPropertyChangeListener(function(e) {
    var node1 = box.getDataById('n1');
    if (e.source === node1 && e.property === 'location') {
        xData.setStyle('chart.value', node1.getX());
        yData.setStyle('chart.value', node1.getY());
    }
});
```

Step6 - charts - BarChart

```
function initBarChart(box) {
    var xData = new twaver.Element();
    var yData = new twaver.Element();
    var chart = new twaver.charts.BarChart();
    chart.getDataBox().add(xData);
    chart.getDataBox().add(yData);
    chart.setYScaleValueGap(25);
    chart.formatYScaleText = function(value) {
         return value.toFixed(0);
    };
    document.body.appendChild(chart.getView());
    chart.getView().style.background = '#E9E9E9';
    chart.adjustBounds({
                  x : 690,
                  v : 170,
                  width : 170,
                  height : 150
              });
    box.addDataPropertyChangeListener(function(e) {
                  var node2 = box.getDataById('n2');
```

}

225	214	
200		
175		
150		
125		
100		86
75		
50		
25		86

```
box.addDataPropertyChangeListener(function(e) {
    var node2 = box.getDataById('n2');
    if (e.source === node2 && e.property === 'location') {
        xData.setStyle('chart.value', node2.getX());
        yData.setStyle('chart.value', node2.getY());
    }
});
```

Step6 - charts - LineChart

```
var chart = new twaver.charts.LineChart();
chart.setValueVisible(false);
chart.setYScaleValueGap(50);
chart.setLowerLimit(0);
chart.setYScaleMinTextVisible(true);
chart.formatYScaleText = function(value) {
    return value.toFixed(0);
}
```

};

```
var e1 = new twaver.Element();
var e2 = new twaver.Element();
var e3 = new twaver.Element();
var e4 = new twaver.Element();
chart.getDataBox().add(e1);
chart.getDataBox().add(e2);
chart.getDataBox().add(e3);
chart.getDataBox().add(e4);
var list1 = new twaver.List();
var list2 = new twaver.List();
var list3 = new twaver.List();
var list4 = new twaver.List();
e1.setStyle('chart.values', list1);
e2.setStyle('chart.values', list2);
e3.setStyle('chart.values', list3);
e4.setStyle('chart.values', list4);
e1.setName('x1');
e2.setName('y1');
e3.setName('x2');
e4.setName('y2');
```

var chartPane = new twaver.charts.ChartPane(chart) document.body.appendChild(chartPane.getView()) chart.getView().style.background = '#E9E9E9'; chartPane.adjustBounds({ x : 690, y : 330, width : 170, height : 150 }); box.addDataPropertyChangeListener(function(e) { var node1 = box.getDataById('n1'); var node2 = box.getDataById('n2'); if (e.property === 'location' && (e.source === node1 || e.source === node2)) { list1.add(node1.getX()); list2.add(node1.getY()); list3.add(node2.getX()); list4.add(node2.getY()); if (list1.size() > 10) { list1.removeAt(0); list2.removeAt(0); list3.removeAt(0); list4.removeAt(0); } chart.invalidateModel(); }

```
});
```

Step6 - charts - ChartPane





Charts


BarChart



PieChart



Step7 - SplitPane, CSS

```
<!DOCTYPE html>
<html>
<head>
    <title>TWaver HTML5 Step by Step</title>
    <link rel="shortcut icon" href="tw.png"/>
    <link href="default.css" rel="stylesheet" type="text/css"/>
    . . .
</head>
<body onload="init()">
    <div id='header'>
        <div>
            <h1>TWaver HTML5 </h1>
            <h2>With step by step, building and practicing the skills
you need</h2>
        </div>
    </div>
    <img id='logo' src='logo.png'/>
</body>
</html>
```

Step7 - SplitPane, CSS



Left

var mainSplit = new twaver.controls.SplitPane(); document.body.appendChild(mainSplit.getView()); var style = mainSplit.getView().style; style.position = 'relative'; style.width = '840px'; style.height = '480px'; style.margin = '0px auto'; style.padding = '0px'; style.background = 'white';

CSS3 & Animate



- border-radius
- text-shadow, box-shadow

}

- transform
- animate



.Node{
 text-align: center;
 color: #880000;
 padding: 10px;
 width: 86px;
 border-radius: 8px;
 box-shadow: 0 0 16px #000000;
 background-color: #88EE11;
 -webkit-transform:rotate(15deg);
 -webkit-transition-property: "-webkit-transform";
 -webkit-transition-duration: 0.35s;
 -webkit-transition-timing-function: ease-in-out;

Step8 - class extends

twaver.Util.ext(subClass, superClass, overrideMethods);

```
// define MyNetwork
var MyNetwork = function (box) {
    MyNetwork.superClass.constructor.call(this, box);
};
twaver.Util.ext('MyNetwork', twaver.network.Network, {
    toString: function(){
        return "MyNetwork";
    }
});
var network = new MyNetwork(box);
```

Step8 - Accordion

var accordion = new twaver.controls.Accordion(); accordion.add('Tree', tree); accordion.add('PropertySheet', sheet); accordion.expand('PropertySheet');

Tree		Tree PropertySheet	
🖃 🗆 🖳 group			
🗹 🛄 node1	N	Name	node1
🖂 🛄 node2		Color	
🗆 🔧 link		Location	x:50, y:50
PropertySheet			

Step8 - TabPane

```
// define MyTabPane
MyTabPane = function (box) {
    MyTabPane.superClass.constructor.call(this);
    this._pieChart = this.createPieChart(box);
    this._barChart = this.createBarChart(box);
    this._lineChart = this.createLineChart(box);
    this.addTab('Line', 'line.png', this._lineChart).setClosable(true);
    this.addTab('Pie', 'pie.png', this._pieChart);
    var tab = this.addTab('Bar', 'bar.png', this._barChart);
    this.getTabBox().getSelectionModel().setSelection(tab);
};
                                                            🔀 Line 💌
                                                                        Pie
                                                                                   Bar
twaver.Util.ext('MyTabPane', twaver.controls.TabPane, {
                                                                     200
    addTab: function (name, icon, chart) {
                                                           200
        var tab = new twaver.Tab(name);
                                                           175
                                                                                 150
        tab.setName(name);
                                                           150
        tab.setIcon(icon);
                                                           125
        tab.setView(new twaver.charts.ChartPane(chart));
                                                           100
        this.getTabBox().add(tab);
                                                           75
        return tab;
                                                            50
    },
                                                           25
});
                                                                        🛛 🗙 📕 y
```

AlarmOverview



Step9 - export xml

twaver.XmlSerializer : function(dataBox, settings, filterFunction) #serialize: function() #deserialize: function(xmlString, rootParent)

```
var serializer = new twaver.XmlSerializer(box);
document.onkeydown = function(evt) {
    if ((evt.ctrlKey || evt.metaKey) &&
    evt.keyCode == 83) {
        evt.preventDefault();
        alert(serializer.serialize());
    }
};
```

<twaver version='1.0' platform='html5'> <dataBox class='MyBox'> <layerBox> <layer visible='true' editable='true' movable='true'/> </layerBox> </dataBox> <data class='twaver.Group' ref='0'> <![CDATA[group]]> true </data> <data class='twaver.Node' ref='1'> <s n='select.color'>yellow</s> <![CDATA[node1]]> </data> <data class='twaver.Link' ref='3'> <s n='select.color'>blue</s> <![CDATA[link]]> </data> </twaver>

TWaver Core Components

- MVC Design Pattern
- Data Model DataBoxs, Elements
- Using Network
- Render Process

MVC Design Pattern



Model & Views 🖃 📃 Node - 8 Child - 7 ID Logo Views(V) Child - 5 SERVA Gro a canal from Child - 0 USA link **3**11 🖃 📃 Node - 6 WPT Power Child - 5 China from ElementUl Element Model(M) **ElementBox** Node Link Group •••

Event-Driven

Element node.setStyle("inner.color", "#FF0000") firePropertyChange("S:inner.color", ...) ElementBox dataPropertyChangeListener _dataPropertyChangeDispatcher.fire(...) handleElementPropertyChange View network, tree, chart ...

ui.handlePropertyChange(...)

Event Listeners

```
//data property change listener
databox.addDataPropertyChangeListener(function(e){
    console.log("data property: '" + e.property + "' changed, old
value: '"
    + e.oldValue + "', new value: '" + e.newValue + "'");
});
```

```
//databox change listener
```

```
databox.addDataBoxChangeListener(function(e){
    if(e.data){
        console.log("'" + e.data.getName() + "' " + e.kind);
    }else{
        console.log("databox " + e.kind);
    }
});
```

Event Listeners

Target	Listener	
DataBox	addDataBoxChangeListener addDataPropertyChangeListener addHierarchyChangeListener addPropertyChangeListener	
ElementBox	extends DataBox addIndexChangeListener	
AlarmBox	extends DataBox	
LayerBox	extends DataBox	
SelectionModel	addSelectionChangeListener	
Network	addInteractionListener addPropertyChangeListener	
AlarmSeverity	addAlarmSeverityChangeListener	

Data & Data Models

Data Item : The basic unit of data

Data Model : the data management container, provides add/ remove/clear operations, can monitor the property change of data

Data Items	Data Models	
Data	DataBox	
Element	ElementBox	
Alarm	AlarmBox	
Layer	LayerBox	

The Relationship Among Data

 ElementBox is a data container for managing network elements, it includes a alarm box, a layer box and a selection model



Using DataBox

DataBox - Data container for managing data items.

- Provides basic operations like add, delete, edit
- Monitors elements changes and other events
- It's the base class for ElementBox, AlarmBox, LayerBox
- It contains selection model

DataBox#getSelectionModel: function ()



Basic Operations

• Add data

add: function (data, index)

• Delete data

remove: function (data)

• Clear data

clear: function ()

Getting Elements

Getting data

getDataByID : function(id)
getDatas : function()
toDatas : function(matchFunction, scope)
getRoots : function()
getSiblings : function(data)

Traverse data

forEach : function(callbackFunction, scope)
forEachByDepthFirst : function(callbackFunction, scope)
forEachByBreadthFirst : function(callbackFunction, scope)

Hierarchy of Elements

- Element hierarchy is built by parent-child relationships
- move***methods of DataBox are used to adjust element index





Quick Finder

QuickFinder:

To find elements by name or other property of element, for example:

var nameFinder = new twaver.QuickFinder(box, "name");
var datas = nameFinder.find("PC");

Create a quick finder:

twaver.QuickFinder = function (dataBox, propertyName, propertyType, valueFunction, filterFunction)

Quick finder types:

- "accessor" find by accessor property
- "client" find by client property
- "style" find by style property

Quick Finder Example

```
var box = new twaver.DataBox();
```

```
var nameFinder = new twaver.QuickFinder(box, "name");
var noFinder = new twaver.QuickFinder(box, "NO", "client");
```

```
for (var i = 0; i < 20; i++) {
    var data = new twaver.Data();
    data.setName("group-" + parseInt(i / 4));
    data.setClient("NO", i % 4);
    box.add(data);
}</pre>
```

```
data = nameFinder.findFirst("group-1");
log("nameFinder.findFirst(\"group-1\")", data);
```

```
var datas = nameFinder.find("group-1");
log("nameFinder.find(\"group-1\")", datas);
```

```
data = noFinder.findFirst(1);
log("noFinder.findFirst(1)", data);
```

```
datas = noFinder.find(1);
log("noFinder.find(1)", datas);
```

<pre>nameFinder.findFirst("group-1")</pre>
name:group-1, NO:0
<pre>nameFinder.find("group-1")</pre>
name:group-1, NO:0
name:group-1, NO:1
name:group-1, NO:2
name:group-1, NO:3
noFinder.findFirst(1)
name:group-0, NO:1
noFinder.find(1)
name:group-0, NO:1
name:group-1, NO:1
name:group-2, NO:1
name:group-3, NO:1
name:group-4, NO:1

Import & Export DataBox

- DataBox can both import and export xml
- The export xml contains properties of all element and DataBox's own properties
- Import and export by XmlSerializer

Import & Export DataBox

XmlSerializer - import & export

twaver.XmlSerializer = function (dataBox, settings, filterFunction)

#serialize()

#deserialize(xmlString, rootParent)

• SerializationSettings - import & export setting

#registerProperty(property, type)
#registerStyle(styleProp, type)
#registerClient(clientProp, type)

DataBox Export Example

```
var box = new twaver.DataBox();
box.setName('box001');
```

```
var data = new twaver.Data();
data.setName('001');
data.setClient('age', 27);
box.add(data);
var child = new twaver.Data();
child.setName('001-1');
child.setParent(data);
box.add(child);
```

```
var settings = new twaver.SerializationSettings();
settings.registerClient('age', "int");
settings.registerProperty('name', "string");
settings.registerProperty("parent", "data");
```

var serializer = new twaver.XmlSerializer(box, settings); console.log(serializer.serialize());

DataBox Import Example

```
box name is box001
                                                    box count is 2
                                                    data name is 001
                                                    age is 27
var xmlString = serializer.serialize();
                                                    data name is 001-1
var newBox = new twaver.DataBox();
                                                    parent is 001
var serializer2 = new twaver.XmlSerializer(newBox, settings);
serializer2.deserialize(xmlString);
console.log('box name is ' + newBox.getName());
console.log('box count is ' + newBox.size());
newBox.forEach(function(data) {
           console.log('data name is ' + data.getName());
           if (data.getClient("age")) {
               console.log('age is ' + data.getClient('age'));
           }
           if (data.getParent()) {
               console.log('parent is ' + data.getParent().getName());
           }
       });
```

Using ElementBox

ElementBox - the data container for topology elements management Inherited from DataBox, retain the functions of DataBox, increase layer box and alarm box twaver.ElementBox# getLayerBox()

getAlarmBox()



Using ElementBox

Increase layerBox、alarmBox

getLayerBox() getAlarmBox()

• Traverse elements by layer

forEachByLayer(callbackFunction, layer)

forEachByLayerReverse(callbackFunction, layer)

• Increase layer change listener

addIndexChangeListener(listener)

Elements

Data



Dummy

- Dummy is invisible on network, but visible on Tree and Table
- Dummy can be used to reorganize tree structure and avoid to affect network



Node

- Basic node type, can set image, outline, render color ...
- twaver.Node is the base class for other elements



Link

• Link, linking two nodes, it has many types and styles, it provides arrows, loop, flowing etc.



Follower

• Follower, it can set a host node, follower will follow it when host is moving.

var NOde = new twaver.Follower(); node.setLocation(100, 100); node.setName('follower'); box.add(node);

```
var host = new twaver.Node();
host.setLocation(140, 140);
node.setHost(host);
host.setName('host');
box.add(host);
```



Group

• Group, contains other nodes, it can close or open, it has many types of shapes


SubNetwork

 Subnetwork, you can drop in or out by double click, when you enter into a subnetwork, the children of the subnetwork will be displayed, Subnetwork has its own background



ShapeNode

ShapeNode, surrounded by a number of control points.



Bus

 Inherited from ShapeNode, connected by a set of control points into line, the links that connected to it are orthogonal distribution



Grid

 Grid, displays child elements in rows and columns, one grid can embed into an other grid, can be used for creating equipment chassis





Grid A



Crid B



Element Properties



Using Network

- Network hierarchy
- Network interactions
- Network filters
- Network generators
- Network auto layout

Network Hierarchy

- -- rootDiv main canvas, contains all topology components
 - -- topDiv – can be used for painting selection rectangle
 - -- attachmentDiv – contains top attachments, like alarm bubble
 - -- layerDiv - contains all nodes and links
 - -- layer n
 - -- layer ...
 - -- default layer
 - -- bottomDiv – bottom canvas, painting shading under the background
 - -- backgroundDiv - color background or image background





Layer Management



var box = network.getElementBox();
var layerBox=box.getLayerBox();

var layerTop=new twaver.Layer("top", 'top');
layerBox.add(layerTop, layerBox.size());

```
var node = new twaver.Node();
box.add(node);
var node2 = new twaver. Node();
box.add(node2);
var link = new twaver.Link(node, node2);
link.setLayerID(layerTop.getId());
link.setName('link at top');
link.setStyle( "link.color", "#000000");
box.add(link);
```



Network Interaction Mode

- Interaction single listener
- Interaction Mode several input handlers combined into an interaction mode

Network#setInteractions(interactions)

 Predefined Modes - Network provides several predefined interaction modes, such as the default mode, edit mode, creating link mode ...

Network#

...

setDefaultInteractions(lazyMode)

setEditInteractions(lazyMode)

setCreateLinkInteractions(linkType)

Custom Interaction

interaction class

```
twaver.Util.ext('highlightInteraction', twaver.network.interaction.BaseInteraction, {
              setUp : function() {
                   this.addListener('mousemove');
              },
              tearDown : function() {
                   this.removeListener('mousemove');
                   this.end();
              },
              handle_mousemove : function(e) {
                   var element = this.network.getElementAt(e);
                   if (element) {
                        this.highlight(element);
                   } else {
                        this.reset();
                   }
               }
                                                                       mouse over, node
         });
                                                                           highlight
setting interaction mode
network.setInteractions([
    new twaver.network.interaction.SelectInteraction(network),
    new twaver.network.interaction.MoveInteraction(network),
    new highlightInteraction(network),
```

new twaver.network.interaction.DefaultInteraction(network)]);

}

```
twaver.Util.ext('highlightInteraction', twaver.network.interaction.BaseInteraction,
         Ł
          setUp : function() {
             this.addListener('mousemove');
          },
          tearDown : function() {
             this.removeListener('mousemove');
             this.end();
          },
          handle_mousemove : function(e) {
             var element = this.network.getElementAt(e);
             if (element) {
                 this.highlight(element);
             } else {
                 this.reset();
             }
          },
             highlightElement : null,
             highlightColor : "#FF0000",
             oldColor : null,
             highlight : function(element) {
                 this.reset();
                 this.oldColor = element.getStyle("inner.color");
                  this.highlightElement = element;
                  element.setStyle("inner.color", this.highlightColor);
             },
             reset : function() {
                 if (this.highlightElement) {
                      this.highlightElement
                               .setStyle('inner.color', this.oldColor);
                  }
             }
        });
```

Network Filters

Filters are used for general control, such as control elements to show or hide, move or not move

Network#
 //visible filter
 get/setVisibleFunction()
 //movable filter
 get/setMovableFunction()
 //editable filter
 get/setEditableFunction()

network.setVisibleFunction(function(element){
 return !(element instanceof twaver.Link);
});

Network Style Functions





Style Function Example

Example for adding prefix "CN-" to element label, and customizing outline color

```
network.setLabelFunction(function(element){
    return "CN-" + element.getName();
});
```

```
network.setOuterColorFunction(function(element){
    return (element instanceof twaver.Node) ? "#FF0000" : "#00FF00" ;
});
```



Network Auto Layouts

• Automatic positioning nodes and links, make them spread out, and showing good.



Default Layouts



Render Process



Render Process - Network



Render Process - List



more

- Unit Test
- Mobile and Touch
- JS Profiler
- Deployment ...

Unit Test

• JSUnit.net

• JS Test Driver

```
var data = new twaver.Data();
var keys = data.getClientProperties();
assertEquals(0, keys.size());
assertEquals(undefined, data.getClient('name'));
data.setClient('name', 'twaver');
assertEquals('twaver', data.getClient('name'));
```

http:// www.jsunit.r	net/runner/tests/jsUnitTestSuite.html Run Stop	
Trace level: no tracin	g 🗧 Close old trace window on new run	Page lo
Status: Running test "	"testTracingWithUndefinedValue"	
Status: Running test " Progress:	"testTracingWithUndefinedValue"	

Mobile & Touch

Viewport, Orientation, Touch

- no mouse but touch
- Touch Event touchstart, touchmove, touchend, touchcancel
- Gesture Event gesturestart, gesturechange, gestureend

<div

ontouchstart="touchStart(event);"
ontouchmove="touchMove(event);" >
</div>

http://developer.apple.com/library/safari/

JS Profiler

- Firebug Profile
- Webkit Profiles
- JSProfiler ycoder.com/jsprofiler/

-		C-11-	B	
Fu	nction	Calls	Percent	
isVisible	Percent(%), Time(ms),	Calls(inv), Avg	(ms). Function	•
insertAfter	⊿ 69.7%, 953 ms,	1 inv, 953 ms.	twaver.network.Network#v	alidate
findFirstElement	⊳ 69.7%, 953 r ⊳ 0%, 0 ms, 2 inv, 0	ms, 1 inv, 953 m ms. <i>twaver.net</i>	is. twaver.network.Network work.Network# fireViewEve	#validateImp ent
getElementUI	29.2%, 400 ms, 1 29.2%, 400 ms, 1	inv, 400 ms. #le 1 inv, 400 ms.	oadDatas #loadElements	
getLayerByElement	▶ 17.2%, 235 m	s, 600 inv, 0.3	ms. twaver.ElementBox#ad	d
	▷ 3.1%, 43 ms, 5 ▷ 2.1%, 29 ms, 1	00 inv, 0 ms. tu 00 inv, 0.2 ms.	aver.Node# setImage twaver.Link# setToNode	
	▷ 1.4%, 20 ms, 1	00 inv, 0.2 ms.	twaver.Link#setFromNode	
	▷ 1.3%, 18 ms, 5	00 inv, 0 ms. two	aver.Data#setName	
		00 inte 0 mm	And Hade Hastland	

Deployment

- UglifyJS <u>github.com/mishoo/UglifyJS2</u>
- YUI Compressor <u>developer.yahoo.com/yui/</u> <u>compressor/</u>

uglifyjs -o demo/twaver.js twaver.js

java -jar yuicompressor-2.4.6.jar twaver.js -o demo/twaver.js





http://dev.w3.org/html5/websockets/

Browser support

IE	Firefox	Chrome	Safari	Safari mobile
10+	6+	14+	5+	iOS 4.2+

http://en.wikipedia.org/wiki/WebSocket

Server support

Jetty	WebSocket- Node	Autobahn Python	cwebsocket	The Server Framework
Java Servlet	Javascript	Python	С	C++ windows

http://en.wikipedia.org/wiki/Comparison_of_WebSocket_implementations

Client

WebSocket

constructor(url, optional protocols)

events -

onopen, onmessage, onclose, onerror

methods -

send(data)
close(optional code, optional reason)

Use

window.WebSocket = window.WebSocket || window.MozWebSocket;

Server - jetty

Homepage - <u>http://www.eclipse.org/jetty</u> Main classes

```
org.eclipse.jetty.websocket.WebSocketServlet
  WebSocket doWebSocketConnect(HttpServletRequest request, String protocol)
org.eclipse.jetty.websocket.WebSocket
org.eclipse.jetty.websocket.WebSocket.OnTextMessage
  void onOpen(Connection connect);
  void onClose(int code, String message);
  void onMessage(String message);
org.eclipse.jetty.websocket.WebSocket.Connection
  void sendMessage(String message);
```

Calling Flow

Client

- - websocket.onopen
 - websocket.onmessage(evt)
 evt.data => json
 init, update ...
 - websocket.send(...)
 - websocket.close()
 - websocket.onclose

Server - jetty

- AlarmServlet#doWebSocketConnect(...) return new AlarmWebSocket();
- AlarmWebSocket#onOpen(connection) sendMessage(...);//init

AlarmWebSocket#onMessage(message) sendMessage to other clients

AlarmWebSocket#onClose(...)

Links



- <u>TWaver HTML5 API Doc</u>
- <u>TWaver HTML5 继承结构和成员说明</u>
- <u>TWaver HTML5 高性能拓扑</u>

• <u>Node.js</u>



- Forum http://twaver.servasoft.com/forum
- Email <u>tw-service@servasoft.com</u>