The Basics

Sample Declaration Markup

```
<ntb:tree id="tree1" cssclass="folders" cssstyle="width:80px;height:400px;"
targetframe="myFrame">
  <ntb:children>
    <ntb:node label="Google" href="http://www.google.ca"></ntb:node>
    <ntb:node label="Tell Me" onclick="alert('what\'s up, doc?')"></
ntb:node>
    <ntb:node label="I have kids">
      <ntb:children>
        <ntb:node label="I am a kid"></ntb:node>
      </ntb:children>
    </ntb:node>
  </ntb:children>
</ntb:tree>
```

Needed for component to work!

Initialization

```
nitobi.loadComponent("mytree");
```

Namespace

```
<html xmlns:ntb="http://www.nitobi.com">
```

Get the Selected Node

To get the selected value you first need to get a reference to the Tree object. To do this you can call nitobi.getComponent(TreeID).

```
var treeObject = nitobi.getComponent('myTree');
```

..or you can use eventArgs in response to an event like onselect:

```
<ntb:tree id="myTree" onselect="myFunction(eventArgs)"></ntb:tree>
```

Then in your JavaScript function:

```
function myFunction(eventArgs) {
  var treeObject = eventArgs.source;
  var selectedNode = treeObject.getSelected()[0]; }
```

Returns an array

So now, in selectedNode we have a reference to the node. We can get the values in that node by accessing the XML using getAttribute(). Tree nodes can have any number of fields that you want to specify, but normally they at least have these fields:

* "key" - the primary key of the row
* "label" - the text of that node
* "nodetype" - 'node' or 'leaf'
* "haschildren" - 'true' or 'false' (boolean)

So in our function, we use getAttribute to specify the field we want:

```
function myFunction(eventArgs) {
  var treeObject = eventArgs.source;
  var selectedNode = treeObject.getSelected()[0];
  var keyValue = selectedNode.getAttribute('key');
  alert(keyValue);   }
```

Events

Events are subscribed to via the declaration. Eg:

```
<ntb:tree onclick="handleClick(eventArgs)" ...>
```

<ntb:tree>

onclick / ondataready / onnodetoggled

<ntb:node>

onclick / ondeselect / onselect

Subscribing to Events with JavaScript

```
var treeObj = nitobi.getComponent("myTree");
treeObj.onClick.subscribe(handleClick);

function handleClick(eA) { }
```

See section on eventArgs for an explanation of this.

Using eventArgs

Many events are able to pass an object to the recipient function that contains transient event information created when the event is triggered. This is achieved by including an eventArgs argument in the declaration for your event. eg:

```
<ntb:tree source="mytb" onclick="myfunct(eventArgs)">
```

Then in your function:

```
function myfunct(eA) { myTabObject = eA.source; }
```

The following properties are available in eventArgs:

```
<function> event() - Returns a bunch of stuff related to the event.
Eg: event.type may equal "click".
<function> source() - Returns the tree object
```

Commonly Used Attributes

<ntb:tree>

cssstyle - String The css style to apply to the tree.
expanded - Boolean Whether or not to automatically expand all nodes initially.
targetframe - String When using frames, tree can be automatically configured to change the URL of the frame based on the value of the node. This is the ID of the frame.
rootenabled - Boolean Whether or not to use a root node (cosmetic)
gethandler - String The url of the gethandler
effect - Animation effect to use. Eg: blind, shade, and none

<ntb:node>

gethandler - String The getHandler for this specific node. Will override any other gethandler specified for the entire tree.
expanded - Boolean Whether or not to automatically expand this node initially.
label / cssclass / haschildren
nodetype - String leaf or node

Important Methods

Important methods for nitobi.tree.Tree

```
find(key, value) - Searches for the given key/
value pair. Returns an array of nodes.
getChildren() - returns the children collection
for a tree.
getRoot() - Returns the root node.
loadData() - Load data from the getHandler.
render() - Repaint the object.
getSelected() - Returns an array of the currently
selected nodes.
```

Important methods for nitobi.tree.Children

```
add(nitobi.tree.Node) - Adds a node to the list of
children.
get(Number) - Returns an item from the IList
insert(Number) - Insert a node at a specific index
remove(Number) - Remove a node from the list of
children.
```

Important methods for nitobi.tree.Node

```
getChildren() - Returns the children collection
for this node.
getDataboundAncestor() - Returns the closest an-
cestor of this node
getParent() - Returns the parent node of this
object.
getTree() - Returns the tree to which this node
belongs.
isLeaf() - Returns true if the node is a leaf
loadData() - Load data from the getHandler
render() - Redraw the node.
```

Inheritance Stuff

```
nitobi.tree.Tree extends nitobi.ui.Element

nitobi.tree.Children extends nitobi.ui.Container

nitobi.tree.Node extends nitobi.ui.Element
```

Tree Effects

Effects through the declaration

```
<ntb:tree effect="blind" … >
```

Valid values for effect are

* 'blind'
* 'shade'
* 'none'

Effects through the API:

```
var myTreeObj = nitobi.getComponent('myTree');
myTreeObj.showEffect = nitobi.effects.families
["shade"].show;
myTreeObj.hideEffect = nitobi.effects.families
["blind"].hide;
```

Sample Use Cases

Get the Tree Object

```
var myTree = nitobi.getComponent('myTree');
```

Get the Children Collection

```
var myCh = myTree.getChildren();
```

Get the 2nd Node in the Collection

```
var secondNode = myCh.get(1);
```

Get the Label of this Node

```
var myLabel = secondNode.getLabel();
```

Expand a Node Programmatically

```
secondNode.loadData();
```

Custom Icon on a Node

Its possible to force a node or leaf to have a custom icon using the icon attribute in the XML.

When using a getHandler, simply define a column called "icon" and make the data the URL for the icon image. Blank fields will be ignored.

When using static data:

```
<ntb:tree id="tree1" cssclass="folders" >
  <ntb:children>
    <ntb:node label="Google" ></ntb:node>
    <ntb:node label="Tell Me" icon="mynode.gif">
    </ntb:node>
    <ntb:node label="I have kids">
      <ntb:children>
        <ntb:node label="I am a kid"></ntb:node>
      </ntb:children>
    </ntb:node>
  </ntb:children>
</ntb:tree>
```

getHandler Attributes

```
get.asp?id=13&label=Australia&nodetype=node&haschildren=true&treeId=tree1
```

id - Tells us the id of the node requesting its children (this will be blank if the tree is requesting its children)
label - The label on the node making the request
nodetype - Will either be "node" or "leaf"
haschildren - true if the node has children, false otherwise
treeId - The id attribute of the Tree declaration

This part is intentionally left blank want to see something else? Let us know:

support@nitobi.com

Troubleshooting

Before losing hope, make sure you check the following things.

Tree Won't Initialize

* Did you define the xmlns:ntb namespace?
* Is your stylesheet/javascript included?
* Did you remember to include nitobi.toolkit.js?
* Remember to call nitobi.loadComponent('myID')?

No Data in Tree

* Datasource is broken? Check using fiddler (http://www.fiddlertool.com) or Firebug (http://www.getfirebug.com)
* Broken declaration.

Troubleshooting Resources

Nitobi Forums - http://forums.nitobi.com

Nitobi Knowledgebase - http://www.nitobi.com/kb/

Email Support - support@nitobi.com

Fiddler Tool - http://www.fiddlertool.com

Firebug - http://www.getfirebug.com