Basic

<?xml version="1.0"?>	
<tree id="0">	top xml tag, mandatory
<item id="p1" text="parent item 1">	each item represented by *item* tag
<item id="c1-0" text="child item 1"/>	
<item id="c1-1" text="child item 2"/>	parent-child relation defined by structure
</item>	
<item id="p2" text="parent item 1"/>	text - caption of tree item
</tree>	

Advanced

<?xml version="1.0"?>	
<tree id="0">	top xml tag, mandatory
<item id="p1" text="parent item 1" select="1" call="1">	pre-selected on load
<userdata name="some1">value1</userdata>	
<item id="c1-0" text="child item 1"/>	
<item id="c1-1" text="child item 2"/>	
</item>	
<item id="p2" text="parent item 1" open="1">	open on load
<userdata name="some1">value2</userdata>	
<item id="c2-0" text="child item 1" im0="a1.gif" im1="a2.gif" im2="a3.gif"/>	with custom images
<item id="c2-1" text="child item 2" tooltip="some tooltip here"/>	
</item>	
<item id="p3">	if you have some complex data as item text
<itemtext>	*itemtext* tag can be used
Some text here	
</itemtext>	
</item>	
<item id="p4" top="1">	
<itemtext><![CDATA[	by using CDATA
any <strong>HTML</strong>	you can set any html here
]]></itemtext>	
</item>	
<item id="p5" text="&lt;em&gt;text&lt;/em&gt;">	HTML also can be used inside text attribute
</item>	
</tree>	

Full

```
<?xml version="1.0" encoding="iso-8859-1"?>                     encoding can be important
<tree id="0" radio="0" order="asc">
 <item id="p1" text="parent item 1" select="1" call="1">        pre-selected on load
  <userdata name="some1">value1</userdata>
  <item id="c1-0" text="child item 1"/>
  <item id="c1-1" text="child item 2"/>
 </item>
 <item id="p2" text="parent item 1" open="1">                   open on load
  <userdata name="some1">value2</userdata>
  <item id="c2-0" text="child item 1" im0="a1.gif" im1="a2.gif"
im2="a3.gif"/>                                                  with custom images
  <item id="c2-1" text="child item 2" tooltip="some tooltip here"/>
 </item>
                                                                if you have some complex data
 <item id="p3">                                                 as item text
  <itemtext>                                                    itemtext tag can be used
   Some text here
  </itemtext>
 </item>
 <item id="p4" top="1">
  <itemtext><![CDATA[                                           by using CDATA
   any <strong>HTML</strong>                                    you can set any html here
  ]]></itemtext>
 </item>
                                                                HTML also can be used inside
 <item id="p5" text="<em<TEXT< span em<>>">                     text attribute
 </item>
 <item id="p6" text="Checkbox related">
                                                                item has radio instead of
  <item id="p6-1" text="some text" radio="1" />                 checkbox
  <item id="p6-2" text="some text" nocheckbox="1" />            item has not checkbox at all
  <item id="p6-a" text="some text" disabled="1" />              item has disabled checkbox
  <item id="p6-b" text="child item 1" checked="1"/>             checked item
  <item id="p-6c" text="child item 2" checked="-1"/>            half-checked item
 </item>
 <item id="p7" text="some text here" child="1" />               item has not loaded childs
 <item id="p8" text="some text" aCol="red" sCol="green" />      item with custom colors
 <item id="p9" text="some text" style="font-weight:bold" />     item with bold text
 <item id="p10" text="some text" closeable="0" />               item can't be closed
 <item id="p11" text="some text" topoffset="30" />              item with custom vertical offset
 <item id="p12" text="some text" locked="1" />                  item will not be accessible
 <item id="p13" text="some text" imwidth="30"
imheight="25"/>                                                 item will image of custom size
</tree>
```

item

tag represent element of tree, the result tree will have same hierarchy as tags in XML

Attributes

aCol	item text in normal state
call	after loading, onSelect handler will be called for item in question independent form selection in tree
checked	if checkboxes enable in tree, based on attribute value checkbox state will be set • 0 or absent - unchecked • 1 - checked • -1 - third state (for 3-state checkboxes)
child	attribute has sense only for dynamical loading in case of dynamical loading - it state that item has sub-items which need to be loaded tree expect that attribute will have value > 0 (specifying exact value of sub-items not necessary, but can be useful in some scenarios)
closeable	if this attribute set to 0 then item will not react on any closing|opening attempts by user action (it will be possible to change its state by JS API)
disabled	if checkboxes used, item will have disabled checkbox
id	identificator of item must be unique per tree can contain any combination of chars|letters
im0	Image of item - "leaf item" (18x18px by default), if missed default image from applied skin will be used
im1	Image of item - "folder in open state" (18x18px by default), if missed default image from applied skin will be used
im2	Image of item - "folder in closed state" (18x18px by default), if missed default image from applied skin will be used
imheight	set height of image for current item expect integer value if you need to set new size for all tree - you can use JS API
imwidth	set width of image for current item expect integer value if you need to set new size for all tree - you can use JS API
locked	items which such attribute not respond on any user actions (item can be unlocked by JS API)
nocheckbox	even if checkboxes enabled, item with such attribute will not have checkbox if you not using checkboxes - attribute not necessary
radio	if checkboxes enabled then checkbox for item in question will be converted to radio button
sCol	item text in selected state
selected	after loading item with such attribute will be preselected
style	any css text here, it will be applied to label of item

text
label which will be rendered on page, text by default, but can contain any combination of HTML tags if them correctly escaped
if attribute skipped - label can be set by itemtext tag

tooltip
tooltip text used by value (need to be enabled by JS api)
if missed - item text used as tooltip

top
really strange attribute
item with such attribute will be rendered as top one in related branch
it will be better to use it with caution

topoffset
allow to set custom space size between current item and previous one
it allow to separate solid tree in few visual groups
component expect that value will be integer (size in pixels)

---

pathend

tag represent start end of path route

Attributes

id
id of path ,
must has pathend tag with equal value
must be unique per tree

---

pathstart

tag represent start point of path route

Attributes

color
Color of line

id
id of path ,
must has pathstart tag with equal value
must be unique per tree

mode
Mode of line : solid, dashed, etc (any value of css "border-type" can be used)

width
Width of line,
integer expected

---

tree

tag represent the whole tree, or some branch inside it

Attributes

id
id of related parent element
by default tree has global root item with ID = 0, it can be redefined inside tree constructor
in case of dynamical loading attribute must point to related parent ID

order
if attribute present , the sorting functionality will be called on load (separate extension!)
none - ignored
1 - ascendant sorting
other value - descedent sorting

radio
if checkboxes enabled in tree, and such attribute present in XML - all checkboxes it tree will be

replaced with radio buttons (with appropriate change in checking logic)

userdata

To store some additional data, which need not to be displayed, but may be useful for some client side operations - userdata sections can be used.

Attributes

name

Because there may be many userdata tags per item, they differ by name attribute. Value of name attribute used on client side to get related tag content.