

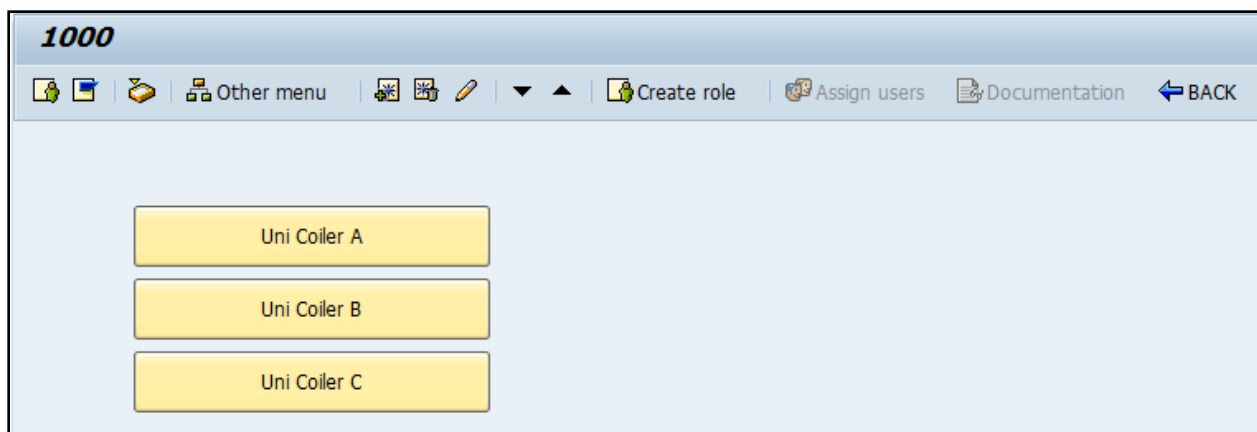
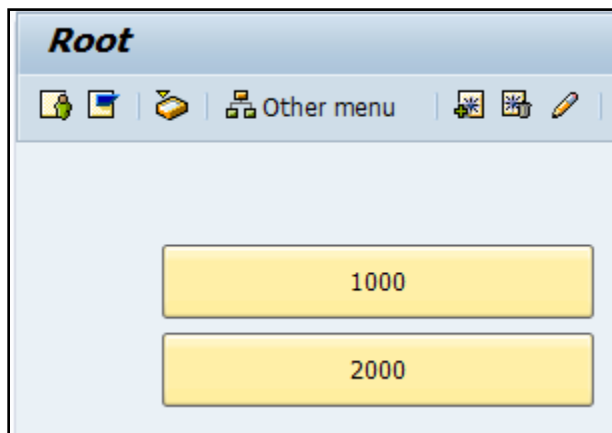
Liquid UI: Dynamic multilevel screens creation based on data

Purpose



Create dynamically multiple levels of hierarchy based on data and pass values along the way to be used in the last level.


User Interface

Log into SAP and on the SAP Easy Access Screen click on the first pushbutton and navigate all the way to land on MM02 screen.








Uni Coiler A





Other menu





Create role

Assign users



Documentation


BACK

Uni Coiler A South East




Uni Coiler A North East



Uni Coiler A South East





Other menu





Create role



Assign users


Documentation

BACK




Uni Coiler A South East 1



Uni Coiler A South East 1





Other menu





Create role

Assign users

Documentation

BACK

Uni Coiler A South East 1A


Uni Coiler A South East 1B

Change Material (Initial Screen)

Select view(s) Organizational levels Data

Material

K1-E-SE-1A



Change Number

Liquid UI Code [Script]

User Interface file –

```
SAPLSMTR_NAVIGATION.ED100.sps
1 // Delete ActiveX Container on SAP Easy Access screen
2 del('X[IMAGE_CONTAINER]');
3
4 // Function to trim blank spaces at the end of the string
5 String.prototype.trim=function(){return this.replace(/^\s+|\s+$/g,'');}
6
7 // Function to check if the string value is blank
8 function isBlank(jvar){
9   if(typeof jvar == 'string') {
10     jvar = jvar.trim();
11   }
12   if(typeof jvar == 'undefined') {
13     jvar = '';
14   }
15   return(jvar == 'undefined' || jvar == undefined || jvar == null || jvar == "" || jvar == void 0);
16 }
17
18 // Function to transform the JSON data into Tree Hierarchical structure
19 function generateHierarchyData(arrName) {
20   // Code to Generate Hierarchy Goes Here
21   // Eventually Return ROOT Node
22   var roots = [], children = {};
23
24   // Find the top level nodes and hash the children based on parent
25   for (var i = 0, len = arrName.length; i < len; ++i) {
26     var item = arrName[i];
27     p = item.Parent;
28     target = !p ? roots : (children[p] || (children[p] = [])); // If isBlank 'p' then target = roots, else target = (children[p] || (children[p] = []))
29     target.push({ value: item });
30   }
31
32   // Function to recursively build the tree
33   var findChildren = function(parent) {
34     if (children[parent.value.Id]) {
35       parent.children = children[parent.value.Id];
36       for (var i = 0, len = parent.children.length; i < len; ++i) {
37         findChildren(parent.children[i]);
38       }
39     }
40   };
41
42   // Enumerate through to handle the case where there are multiple roots
43   for (var i = 0, len = roots.length; i < len; ++i) {
44     findChildren(roots[i]);
45   }
46   return roots;
47 }
```

```

49 // Function to create the UI based on parameters passed
50 function generateHierarchyUI(param) {
51     nodeId = param.currentNodeId;
52     nodeVal = param.currentNodeVal;
53     arrName = param.arrNm;
54     children = [];
55
56     for(var i=0; i<arrName.length; i++){ // For display for Children and Title
57         var item = arrName[i];
58         if(item.Id == nodeId){ // To Display Title
59             if(isBlank(txcode)){
60                 txcode = item.Txcode;
61             }
62             titleName = item.Name;
63             title(titleName); // Push button that is clicked is displayed as the title
64         }
65         if(item.Parent == nodeId){ // Create children array
66             children.push({value: item});
67         }
68     }
69
70     onscreen 'SAPLSMTR_NAVIGATION.0100'
71     pushScreens = false;
72     enter('?');
73 }
74
75 // Function to perform some action on the last level
76 // Navigates to the transaction code and uses the value passed
77 function navigateNode(param) {
78     var tcode = param.l_tcode;
79     first_time = '';
80     enter('/n'+tcode);
81
82     onscreen '*'
83     title(_title);
84     if(tcode == 'MM02'){ // Rest of the navigation logic based on Transaction code
85         set('F[Material]',param.nodevalue);
86     }
87
88     enter('?');
89 }
90
91 // Function to display the previous screen when Back button is clicked
92 function generateHierarchyUIBack(){
93     backClicked = 'X';
94     screenArr.pop();
95     titles.pop();
96     children = screenArr[screenArr.length -1].slice(0);
97 }
98

```

```

99 // Retrieve Data in the below format
100 // These data may come from-
101 // a. Fixed File
102 // b. RFC Call
103 var testData = [
104   {"Id": "1", "Name": "1000", "Value": "", "Parent": "", "Txcode": "MM02"},
105   {"Id": "2", "Name": "Uni Coiler A", "Value": "K1-A", "Parent": "1"},
106   {"Id": "3", "Name": "Uni Coiler B", "Value": "K1-B", "Parent": "1"},
107   {"Id": "4", "Name": "Uni Coiler C", "Value": "K1-C", "Parent": "1"},
108   {"Id": "5", "Name": "Uni Coiler A South East", "Value": "K1-E-SE", "Parent": "2"},
109   {"Id": "6", "Name": "Uni Coiler A North East", "Value": "K1-E-NE", "Parent": "2"},
110   {"Id": "7", "Name": "Uni Coiler A South East 1", "Value": "K1-E-SE-1", "Parent": "5"},
111   {"Id": "8", "Name": "2000", "Value": "", "Parent": "", "Txcode": "VA02"},
112   {"Id": "9", "Name": "Uni Coiler A South East 1A", "Value": "K1-E-SE-1A", "Parent": "7"},
113   {"Id": "10", "Name": "Uni Coiler A South East 1B", "Value": "K1-E-SE-1B", "Parent": "7"}
114 ];
115
116
117 // Initial Entry - SAP Easy access
118 if(isBlank(first_time)){ // UI for Initial entry on Log on
119   first_time = "X";
120   txcode = "";
121   counter = 0;
122   root_val = [];
123   root_val = generateHierarchyData(testData); // Pass the Data Array as parameter to the function
124   firstTitle = 'Root';
125   title(firstTitle); // Initial Title to be displayed
126   for(i=0;i<root_val.length;i++){
127     pushbutton([(i+1)*2,10], ""+root_val[i].value.Name+"", {"size":[2,30], "process":generateHierarchyUI,
128       "using":{"currentNodeId":root_val[i].value.Id, "currentNodeVal":root_val[i].value.Value, "arrNm":testData}});
129   }
130   screenArr = [];
131   titles = [];
132   screenArr.push(root_val);
133   titles.push(firstTitle);
134 } else{ // UI for Second level onwards
135   if(isBlank(children)){ // If user refreshes the screen OR On reaching the last level
136     if(!isBlank(txcode)){
137       enter(("process":navigateNode, "using":{"l_tcode":txcode, "nodevalue":nodeVal}));
138     } else{
139       enter('/O');
140       first_time = '';
141     }
142     goto SCRIPT_END;
143   } else { // For display Multiple levels
144     if(children.length != 0){
145       for(i=0;i<children.length;i++){
146         pushbutton([(i+1)*2,10], ""+children[i].value.Name+"", {"size":[2,30], "process":generateHierarchyUI,
147           "using":{"currentNodeId":children[i].value.Id, "currentNodeVal":children[i].value.Value, "arrNm":testData}});
148       }

```

```

149 // Code for BACK button below
150 if(backClicked == 'X'){
151   titleName = titles[titles.length -1];
152   backClicked = '';
153 }
154 title(titleName);
155 if(titleName != firstTitle){
156   pushbutton([TOOLBAR], "@9S@BACK", '?', {"process":generateHierarchyUIBack});
157 }
158
159 if(!pushScreens){
160   screenArr.push(children);
161   titles.push(titleName);
162   pushScreens = true;
163 }
164 }
165 }
166 }
167
168 SCRIPT_END;;

```