021. How-to initially expand a defined set of tree nodes in an ADF bound ADF Faces tree

Abstract:

ADF Faces trees usually render with their nodes in discosed state. Some use cases however require the tree to render with a defined set of nodes to be initially expanded.

This how-to article explains how to expand the first level nodes of an ADF bound tree upon initial page rendering.

Author: Frank Nimphius, Oracle Corporation
twitter.com/fnimphiu
23-AUG-2010
Introduction

The ADF Faces tree component has a disclosed rowKeySet property that can be used to programmatically set the node keys that should display in disclosed state when the tree is refreshed or the view initially renders. In the Oracle JDeveloper 11g R1 PS2 sample for this article, the initial state is programmatically defined so that all first level nodes are disclosed.
How-to initially expand a defined set of tree nodes in an ADF bound ADF Faces tree

The same sample used with a treeTable is shown below. The tree table sample also checks if the RowKeySet has entries and if it doesn’t set the default.

The tree data in the sample is queried from an ADF Business Components business service through the ADF binding layer. A phase listener on the enclosing view is used to ensure the row keys for the disclosed nodes is set before rendering. The initial page renders as shown in the image above.

Defining the PhaseListener for the f:view component

In the sample application, all tree nodes of the first tree level are expanded. For this, in a phase listener, the ADF Faces tree component is looked up in the view and the JUCtrlHierBinding reference to the ADF binding layer is obtained through the CollectionModel read from the tree value property. With the ADF tree binding handle, the root node of type JUCtrlHierNodeBinding is accessed and used to determine all the level children. In an iteration, the rowKeys of the child nodes are read and added to a java.util.List, which then is added as an entry to the RowKeySetImpl class that represents the disclosed RowKeySet of the tree component.

```java
import javax.faces.component.UIComponent;
import javax.faces.component.UIViewRoot;
import javax.faces.context.FacesContext;
import javax.faces.event.PhaseEvent;
import javax.faces.event.PhaseId;
import oracle.adf.view.rich.component.rich.data.RichTree;
import oracle.adf.view.rich.context.AdfFacesContext;
```
import oracle.jbo.uicli.binding.JUCtrlHierBinding;
import oracle.jbo.uicli.binding.JUCtrlHierNodeBinding;

import org.apache.myfaces.trinidad.model.CollectionModel;
import org.apache.myfaces.trinidad.model.RowKeySet;
import org.apache.myfaces.trinidad.model.RowKeySetImpl;

... public void beforeRenderOnViewLoad(PhaseEvent phaseEvent) {
    if (phaseEvent.getPhaseId() == PhaseId.RENDER_RESPONSE){
        FacesContext fctx = FacesContext.getCurrentInstance();
        AdfFacesContext adfFacesContext =
            AdfFacesContext.getCurrentInstance();

        //make sure this code is executed on the initial page request only
        boolean isInitialRequest = !adfFacesContext.isPartialRequest(fctx);
        if (isInitialRequest) {
            UIViewRoot viewRoot = fctx.getViewRoot();
            UIComponent tree = viewRoot.findComponent("tree1");
            if (tree != null) {
                CollectionModel model =
                    (CollectionModel)((RichTree)tree).getValue();
                JUCtrlHierBinding treeBinding =
                    (JUCtrlHierBinding)model.getWrappedData();
                JUCtrlHierNodeBinding rootNode =
                    treeBinding.getRootNodeBinding();
                RowKeySet rks = ((RichTree)tree).getDisclosedRowKeys();
                if (rks == null) {
                    rks = new RowKeySetImpl();
                }
                List<JUCtrlHierNodeBinding> firstLevelChildren =
                    rootNode.getChildren();
                for (JUCtrlHierNodeBinding node : firstLevelChildren) {
                    ArrayList l = new ArrayList();
                    l.add(node.getRowKey());
                    rks.add(l);
                }
                ((RichTree)tree).setDisclosedRowKeys(rks);
            }
        }
    }
}

The phase listener itself is referenced in a managed bean from the "beforePhase" method property of the fview component.
How-to initially expand a defined set of tree nodes in an ADF bound ADF Faces tree

```java
<f:view beforePhase="# {treeSampleBacking.beforeRenderOnViewLoad}">
    ...
</f:view>
```

### RELATED DOCUMENTATION

- [ ]
- [ ]
- [ ]
- [ ]