What's new in Telosys 1.0.1 ?

1/ Telosys-UIL : Nested loops in TagLib

Nested loops are now supported by the Telosys TagLib

In order to manage nested loops, each loop must be identified.
So a "**loopname**" attribute has been added to loop tags.

If the loop iterates on a collection, using an item object for each collection, the item must be identified to avoid crushing. So an "**itemname**" attribute has been added to loop tags

New attributes by tag :

- **iterate**
  - loopname
  - itemname

- **listbody**
  - loopname
  - itemname

- **loop** (no item name because no collection)
  - loopname

- **comboitems** (no loopname because there's no reason to nested a loop in comboitems)
  - itemname

- **listitems** (no loopname because there's no reason to nested a loop in listitems)
  - itemname

For examples, see the end of this document.

2/ Telosys-DAL : ConnectionProvider custom class

It's now possible to define a specific class as a ConnectionProvider (idea submitted via a patch in the forum). The custom class is specified in the "dbconfig" file.

dbconfig.xml example :
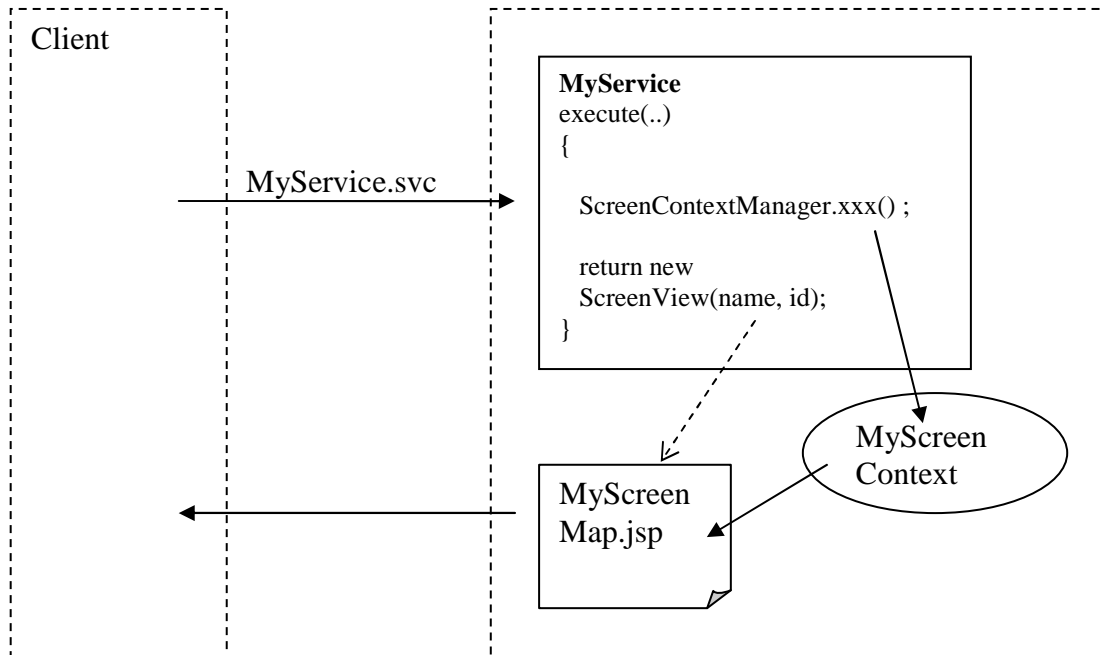
```
<db id = "2"
      name  = "Custom class MyConnectionProvider"
      customclass = "demo.env.MyConnectionProvider" />
```

The custom class must implement the "**ConnectionProvider**" interface.

3/ Telosys-SVC : ScreenContex accessible in the screenmap produced by a service

The ScreenMap produced by a navigation service (a service returning a "ScreenView") can access its associated ScreenContext if any.
So it's possible to initialize the fields with values coming from ScreenContext attributes or ScreenData, like in a classical MVC model.

4/ Telosys-JS : FileUpload : customizable alert message if the upload failed

When an error occurs during an upload …
- If there's a callback function (defined in the application), this callback function is called with the error code and error message as parameters.
- If there's no callback function, the default error message is printed using a "alert" message box

(before this release, the default message was always printed when an error occurred, and the only workaround was to override the fwkUploadCallBack function)

Nested loops example 1 :

```
<t:set y="80" ygap="40" />
<t:loop from="1"  to="6" loopname="loop1" >

 <t:label x="40" txt="In loop @{#count} : " />

 <t:field x="140" value="@{#count}" maxlength="4" />

 <t:combobox x="300" value="@{#count}" >
  <t:loop from="1"  to="5" loopname="loop2" >
   <t:comboitem value="@{#count}" txt="@{#count} - aaa "/>
  </t:loop>
 </t:combobox>

 <t:set y="++" />
</t:loop>
```

Nested loops example 2 :

```
<t:listbody id="myListBody1" loop="1:8" height="240" >
<t:listrow  onclick="alert('Click on ROW : count = @{#count} ')" >
 <t:listcell><t:count/></t:listcell>
 <t:listcell><t:index/></t:listcell>
 <t:listcell>

   <t:listbody id="sublist@{#count}" data="MYITEMS" loopname="nested1"
itemname="item" >
     <t:listrow  selectable="false" >
       <t:listcell width="60" ><t:count/></t:listcell>
       <t:listcell width="40" ><t:out v="@{item.1}"/> </t:listcell>
       <t:listcell width="40" ><t:out v="@{item.2}"/> </t:listcell>
     </t:listrow >
   </t:listbody >

 </t:listcell>
</t:listrow>
</t:listbody >
```