HOW to transform a WComp bean to a NODE RED flow

1. Discovery of WComp

to know more about WCOMP, You can refer to the documentation available online as well as to the demonstration videos available at the following address for the installation and the handling of the WComp environment:

http://rainbow.i3s.unice. fr/wikiwcomp/doku.php?id=download_telechargement

1. Creating a Bean Component

Let's take the example that follows. If we change the TextBox 1 value, the TextBox 2 value will be changed automatically, and if we change the TextBox 3 the value of text box 2 will be overridden. While generating JSON files to create NODE RED flow.

First we have creates a new Bean.cs file (which is renamed to TextBox.cs) All you have to do is to modify the code skeleton on the IDE visual studio to give this new component the desired behaviour and `generate a NODE RED template flow from json file.`

```
1 référence
public TextBox(string flow_name, NodeRedAPI.NodeRedAPI api) : base(flow_name, api)
{
    this.flow_name = flow_name;
    this.AddTextBoxFlow();
}

1 référence
public dynamic genarateJason()
{
    string jsonfile = "../../BeanToFlow/BasicUI/test2.js";
    string Clear_id = "";
    string Set_Text_id = "";
    string TextChanged_id = "";
    string EventGetPropRet_id = "";
    string GetProp_id = "";
    string link_in_Clear_id = "";
    string link_in_Set_Text_id = "";
    string link_in_GetProp_id = "";
    string link_out_TextChanged_id = "";
    string link_out_EventGetPropRet_id = "";
    string ui_text_input_id = "";
```

2. Create a NODE RED flow

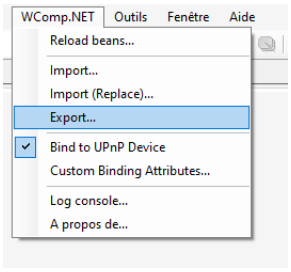
We have implement the class Operation.cs that will use a template jason file(test2.js) to create a nodered flow.

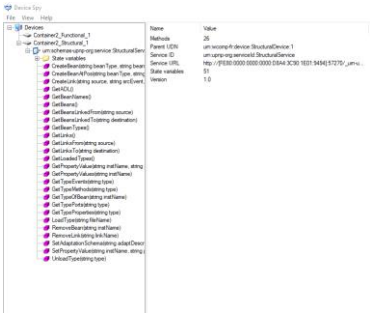
```
1 référence
public void getFlows()
{
    //UPnPArgument uPnPArgument = new UPnPArgument("flowname", "590db922.6d19c8");
    UPnPArgument[] uPnPArguments = new UPnPArgument[1];
    //uPnPArguments[0] = uPnPArgument;
    UPnPAction action = this.service.GetAction("GetFlows");
    //Console.WriteLine(action.ArgumentList.Length);
    this.service.InvokeSync("GetFlows", null);
    this.service.GetActions();

}


0 références
private void AddNode()
{
    UPnPAction[] noderedActions = this.service.GetActions();
    Console.WriteLine(noderedActions[0].ArgumentList);
    var arguments = new UPnPArgument[2];
    arguments[0] = new UPnPArgument("node", "1");
    arguments[1] = new UPnPArgument("flowId", "f32eab8b.153d28");
    this.service.InvokeAsync("AddNode", arguments);

}
}
}
```

3. UPnP Wizard Designer

This tool is like the Intel Device Spy application. It allows to detect the presence and absence of UPnP service. In addition, each time the tool discovers a new UPnP service, it sends a command to the WComp container's the tool is connected with,to to construct a new proxy component corresponding to this new UPnP service.

To test it, after starting WComp and the UPnP Designer, activate the connection (Bind to UPnP Device on WComp) and start a new UPnP device.





For our bean, to discover devices with UPNP we have to implemente a code (in our case is **DiscoverDevice.cs**). this class found wcomp device « um:wcomp - fr:device: StructuralDevice: 1 ». then it checks the statevariable(output) in the service, when u do something(add bean,link) in wcomp , something will appear in the output variable(value)

```
1 référence
private void StateValueChanged(UPnPStateVariable sender, object newvalue)
{
    Console.WriteLine("output: "+sender.Value);
    CheckStateVariable csv = new CheckStateVariable(this.actions);

    Operation op = new Operation(api, csv.CheckValue(sender.Value.ToString()));
}
```

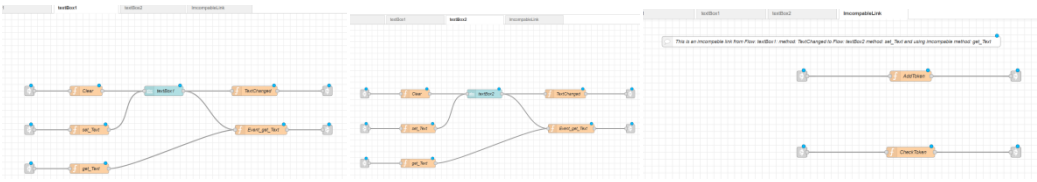
4. create an assembly for the implementation of the application

Make a first assembly that will show the synchronisation between textBox 1 and textBox 2.
Let's start with :

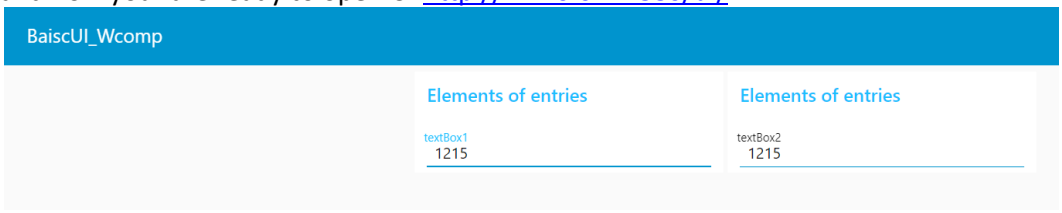
- open nodered
- open wcomp
- bind wcomp to upnp
- open device spy
- click service: subscibe to events
- run the project in vs

Next,

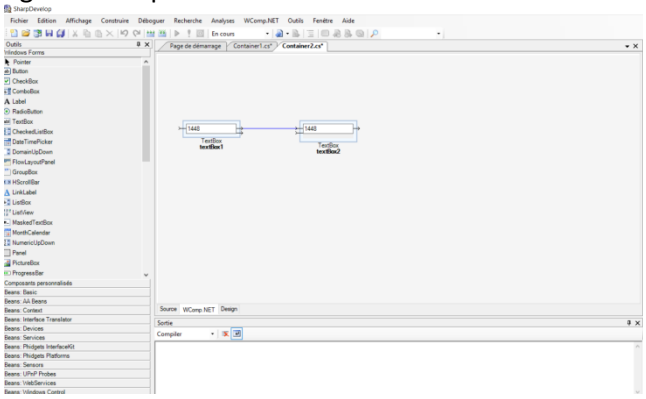
- add a textbox1 in wcomp
- add a text box2 in w comp
- add an incompatible link between textBox1 and textBox2
- click on the link : choose textchanged in tab1, choose incompatble link, choose set_text in tab2, then get_text
- refresh the NODE RED page and you will see an incompatble link



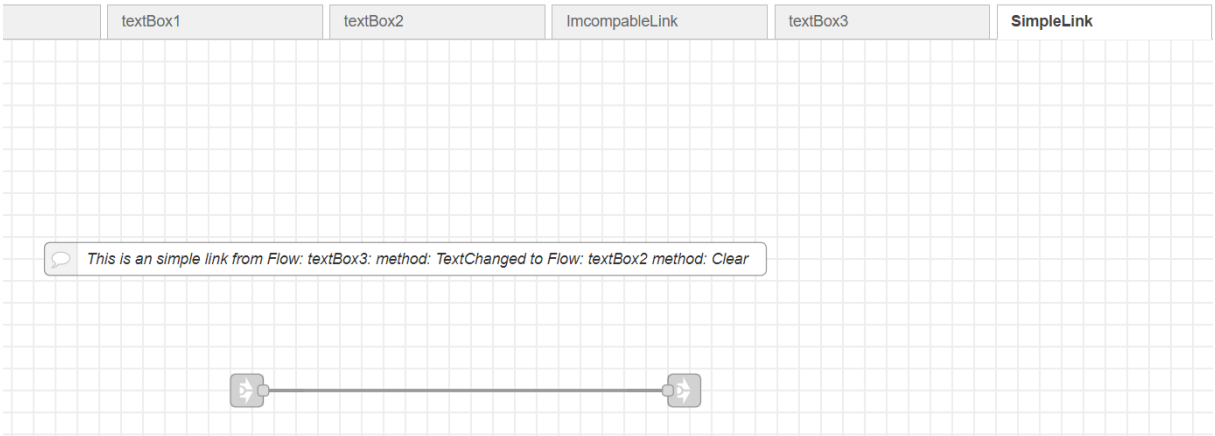
- and now your are ready to open UI http://127.0.0.1:1880/ui/



And the same thing in WComp



- Now add a textbox3 in wcomp
- Add a simple link between textBox3 and textBox2
- Choose textchanged in tab1 and void_clear() in tab2
- Refresh NODE RED, a simple link will be created



- Refresh NODE RED ui