Add Signals to your Layout with JMRI/PanelPro

Dick Bronson - *RR-CirKits, Inc.*

Other Clinics in this series:

- Introduction to Layout Control with JMRI/PanelPro

  8:30 PM, Sunday, July 13th

- Create a Detailed CTC Machine Model with JMRI/PanelPro

  10:00 PM, Monday, July 14th

- Introduction to Layout Control with JMRI/PanelPro

  Repeated 4:00 PM, Friday, July 18th

XML Editing

- At the end of Intro to PanelPro Clinic 3 we promised to cover manual editing of the Panel.xml files.
  - Select an editor that will NOT reformat your files.
  - Select an editor that does NOT think it automatically knows how to edit any file with a .xml extension.
  - For Windows users 'XML Marker' available from http://symbolclick.com/ is an excellent option.
  - On Linux I am using Bluefish which is a standard programming editor with a cute logo.

- Where we are going:
  - SSL – Simple Signal Logic (SSL-clinic-2)

XML Editing

Manual Editing

First copy the file named PanelEditorClinic3.xml and save a new copy with the name SSLClinic1.xml

Open SSLClinic1.xml with your editor.

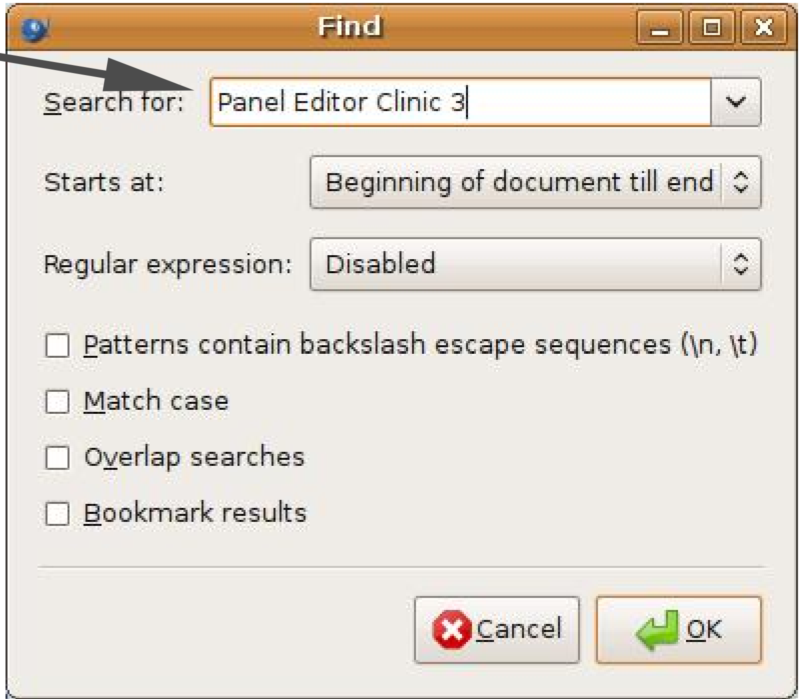
XML Editing

Manual Editing

First copy the file named PanelEditorClinic3.xml and save a new copy with the name SSLClinic1.xml

Open SSLClinic1.xml with your editor.

Find the text ”Panel Editor Clinic 3” so we can change the panel name.



Find

Search for: Panel Editor Clinic 3

Starts at: Beginning of document till end

Regular expression: Disabled

□ Patterns contain backslash escape sequences (\n, \t)

□ Match case

□ Overlap searches

□ Bookmark results

XML Editing

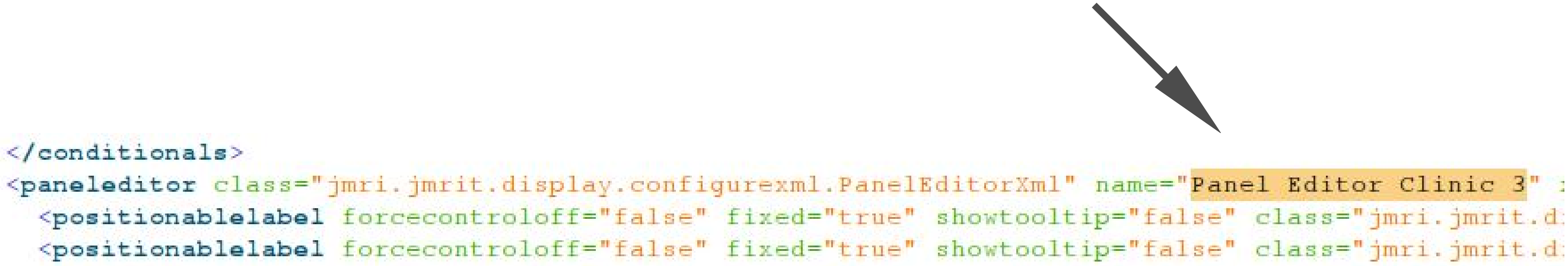
Manual Editing

First copy the file named PanelEditorClinic3.xml and save a new copy with the name SSLClinic1.xml

Open SSLClinic1.xml with your editor.

Find the text ”Panel Editor Clinic 3” so we can change the panel name.

Change the text to read ”SSL Clinic 1”.



```
</conditionals>
<paneleditor class="jmri.jmrit.display.configurexml.PanelEditorXml" name="Panel Editor Clinic 3"
  <positionablelabel forcecontroloff="false" fixed="true" showtooltip="false" class="jmri.jmrit.d
  <positionablelabel forcecontroloff="false" fixed="true" showtooltip="false" class="jmri.jmrit.d
```

XML Editing

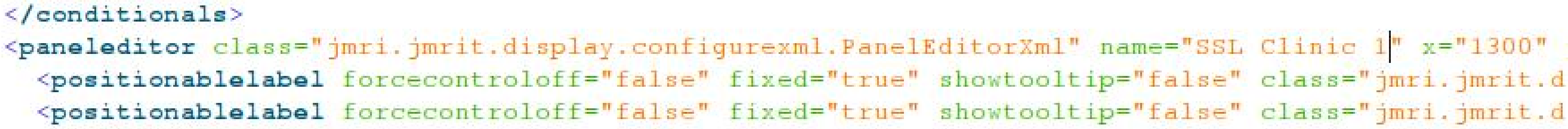
Manual Editing

First copy the file named PanelEditorClinic3.xml and save a new copy with the name SSLClinic1.xml

Open SSLClinic1.xml with your editor.

Find the text ”Panel Editor Clinic 3” so we can change the panel name.

Change the text to read ”SSL Clinic 1”.



```
</conditionals>
<paneleditor class="jmri.jmrit.display.configurexml.PanelEditorXml" name="SSL Clinic 1" x="1300"
  <positionablelabel forcecontroloff="false" fixed="true" showtooltip="false" class="jmri.jmrit.d
  <positionablelabel forcecontroloff="false" fixed="true" showtooltip="false" class="jmri.jmrit.d
```

XML Editing

Manual Editing

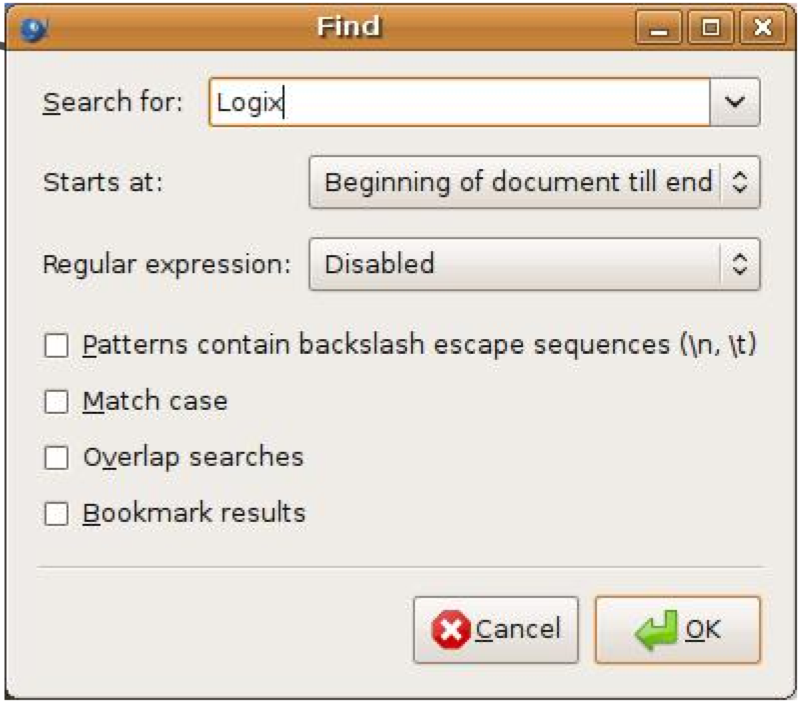
What we need to do is duplicate the Logix used for controlling LT1 (at position 5) for the other three turnout controls.

XML Editing

Manual Editing

What we need to do is duplicate the Logix used for controlling LT1 (at position 5) for the other three turnout controls.

Search for ”Logix”



Search for: Logix

Starts at: Beginning of document till end

Regular expression: Disabled

☐ Patterns contain backslash escape sequences (\n, \t)

☐ Match case

☐ Overlap searches

☐ Bookmark results

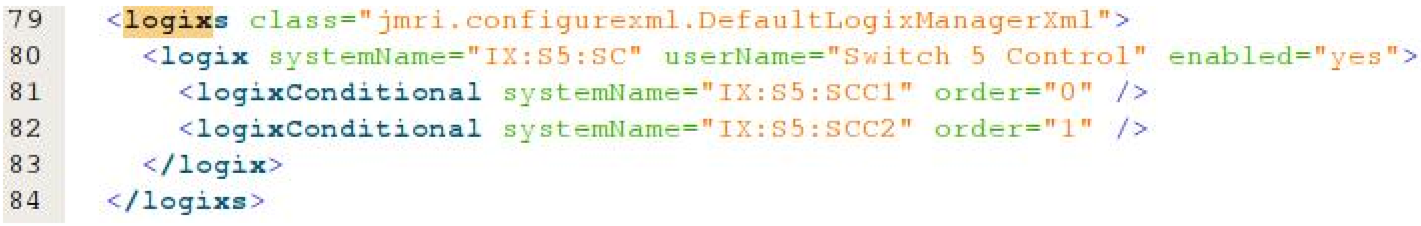
XML Editing

Manual Editing

What we need to do is duplicate the Logix used for controlling LT1 (at position 5) for the other three turnout controls.

Search for ”Logix”

This brings us to the section of the file where the Logix and Conditionals are defined.



```
<logixs class="jmri.configurexml.DefaultLogixManagerXml">
  <logix systemName="IX:S5:SC" userName="Switch 5 Control" enabled="yes">
    <logixConditional systemName="IX:S5:SCC1" order="0" />
    <logixConditional systemName="IX:S5:SCC2" order="1" />
  </logix>
</logixs>
```

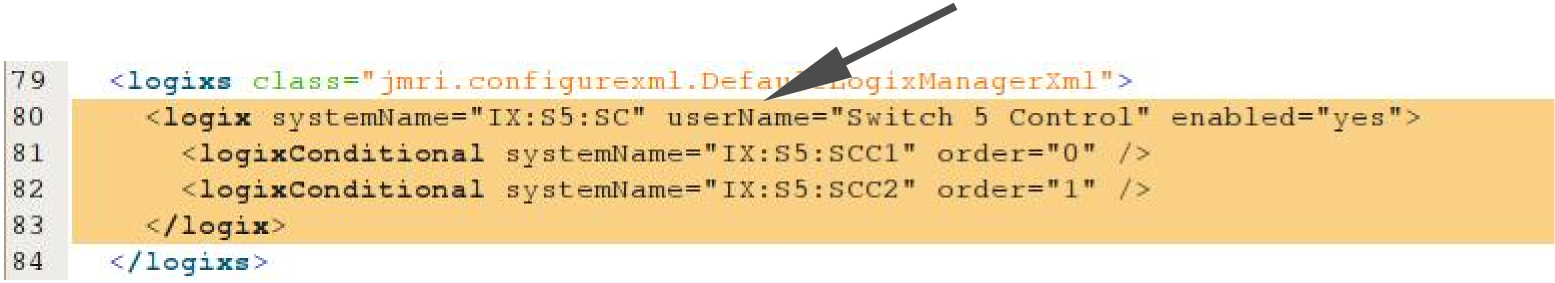
Manual Editing

What we need to do is duplicate the Logix used for controlling LT1 (at position 5) for the other three turnout controls.

Search for ”Logix”

This brings us to the section of the file where the Logix and Conditionals are defined.

Highlight the section starting with ”<logix ” and ending with line ”</logix>”.

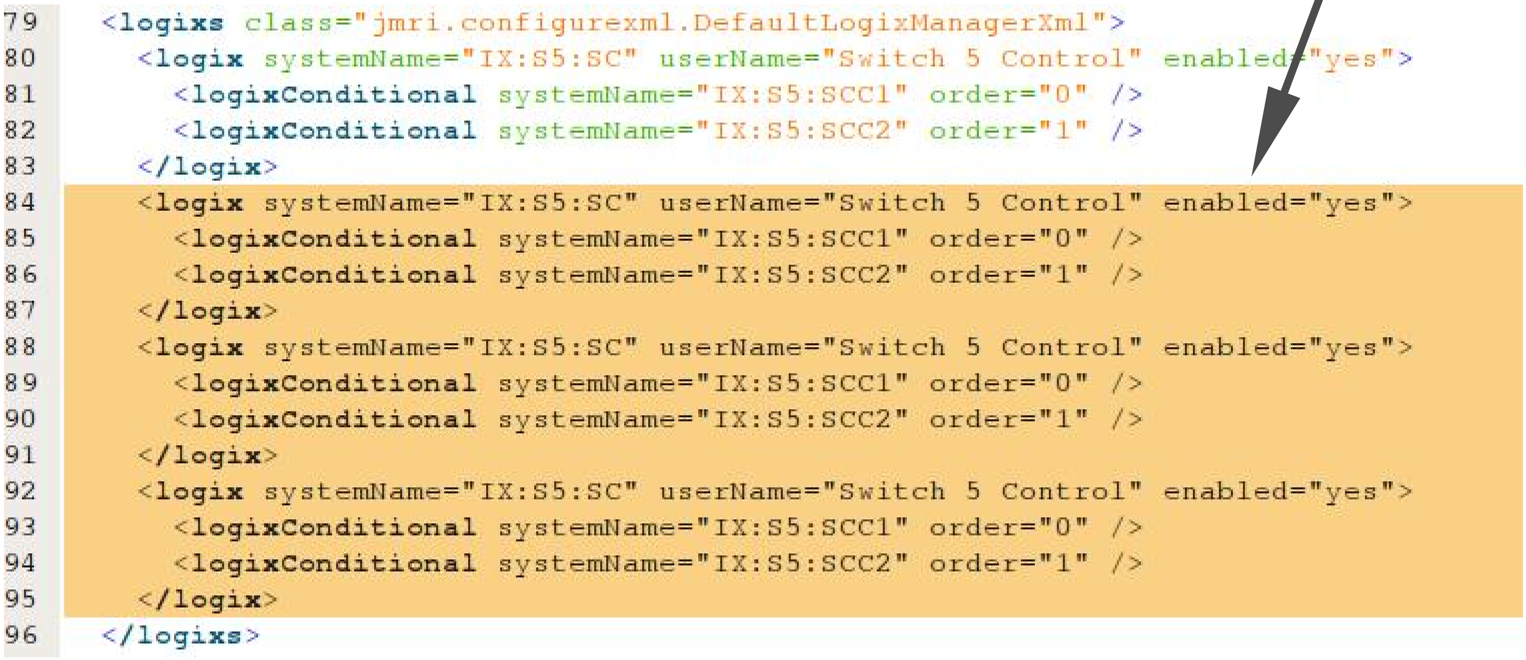


```
<logixs class="jmri.configurexml.DefaultLogixManagerXml">
  <logix systemName="IX:S5:SC" userName="Switch 5 Control" enabled="yes">
    <logixConditional systemName="IX:S5:SCC1" order="0" />
    <logixConditional systemName="IX:S5:SCC2" order="1" />
  </logix>
</logixs>
```

XML Editing

Manual Editing

Copy and then Paste three times to get three added entries that we can edit to match the turnouts.



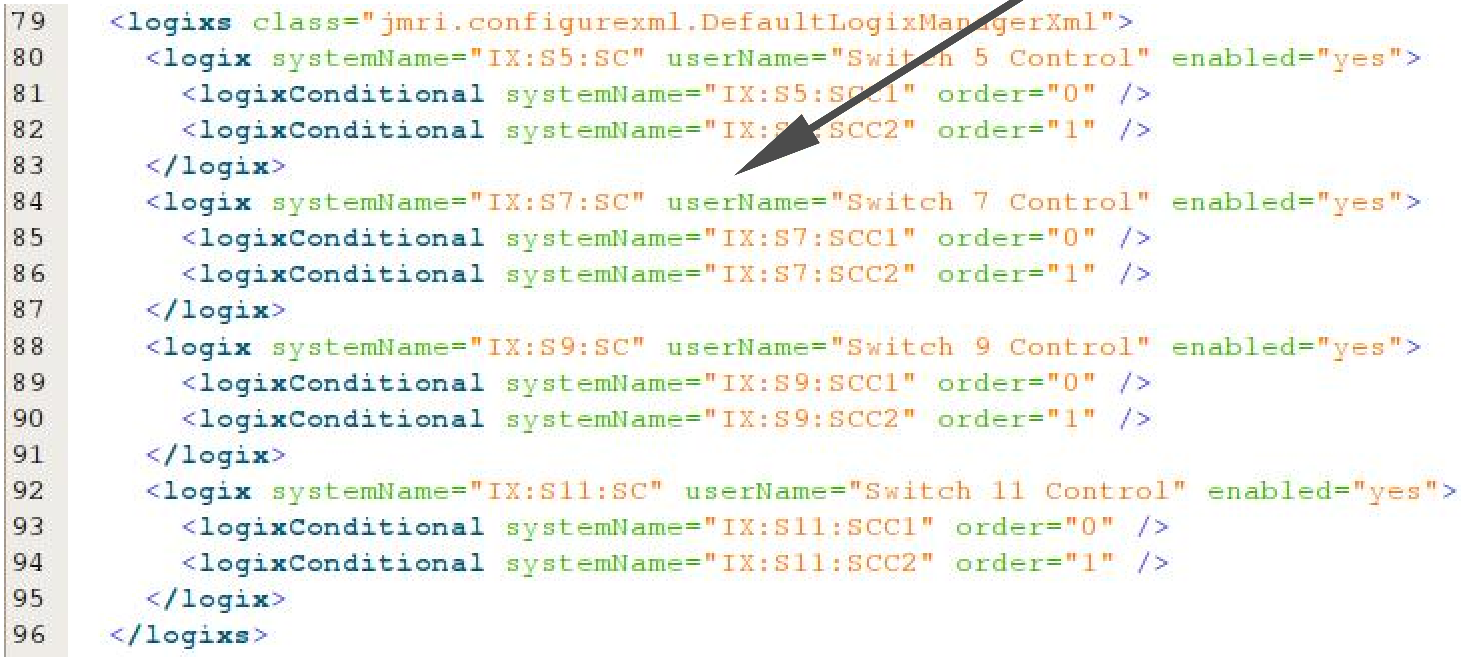
```
<logixs class="jmri.configurexml.DefaultLogixManagerXml">
  <logix systemName="IX:S5:SC" userName="Switch 5 Control" enabled="yes">
    <logixConditional systemName="IX:S5:SCC1" order="0" />
    <logixConditional systemName="IX:S5:SCC2" order="1" />
  </logix>
  <logix systemName="IX:S5:SC" userName="Switch 5 Control" enabled="yes">
    <logixConditional systemName="IX:S5:SCC1" order="0" />
    <logixConditional systemName="IX:S5:SCC2" order="1" />
  </logix>
  <logix systemName="IX:S5:SC" userName="Switch 5 Control" enabled="yes">
    <logixConditional systemName="IX:S5:SCC1" order="0" />
    <logixConditional systemName="IX:S5:SCC2" order="1" />
  </logix>
  <logix systemName="IX:S5:SC" userName="Switch 5 Control" enabled="yes">
    <logixConditional systemName="IX:S5:SCC1" order="0" />
    <logixConditional systemName="IX:S5:SCC2" order="1" />
  </logix>
</logixs>
```

XML Editing

Manual Editing

Copy and then Paste three times to get three added entries that we can edit to match the turnouts.

Change each 'systemName' and 'userName' to match the new added positions.   (12 places)



```
<logixs class="jmri.configurexml.DefaultLogixManagerXml">
  <logix systemName="IX:S5:SC" userName="Switch 5 Control" enabled="yes">
    <logixConditional systemName="IX:S5:SCC1" order="0" />
    <logixConditional systemName="IX:S5:SCC2" order="1" />
  </logix>
  <logix systemName="IX:S7:SC" userName="Switch 7 Control" enabled="yes">
    <logixConditional systemName="IX:S7:SCC1" order="0" />
    <logixConditional systemName="IX:S7:SCC2" order="1" />
  </logix>
  <logix systemName="IX:S9:SC" userName="Switch 9 Control" enabled="yes">
    <logixConditional systemName="IX:S9:SCC1" order="0" />
    <logixConditional systemName="IX:S9:SCC2" order="1" />
  </logix>
  <logix systemName="IX:S11:SC" userName="Switch 11 Control" enabled="yes">
    <logixConditional systemName="IX:S11:SCC1" order="0" />
    <logixConditional systemName="IX:S11:SCC2" order="1" />
  </logix>
</logixs>
```

XML Editing

Manual Editing

Copy and then Paste three times to get three added entries that we can edit to match the turnouts.

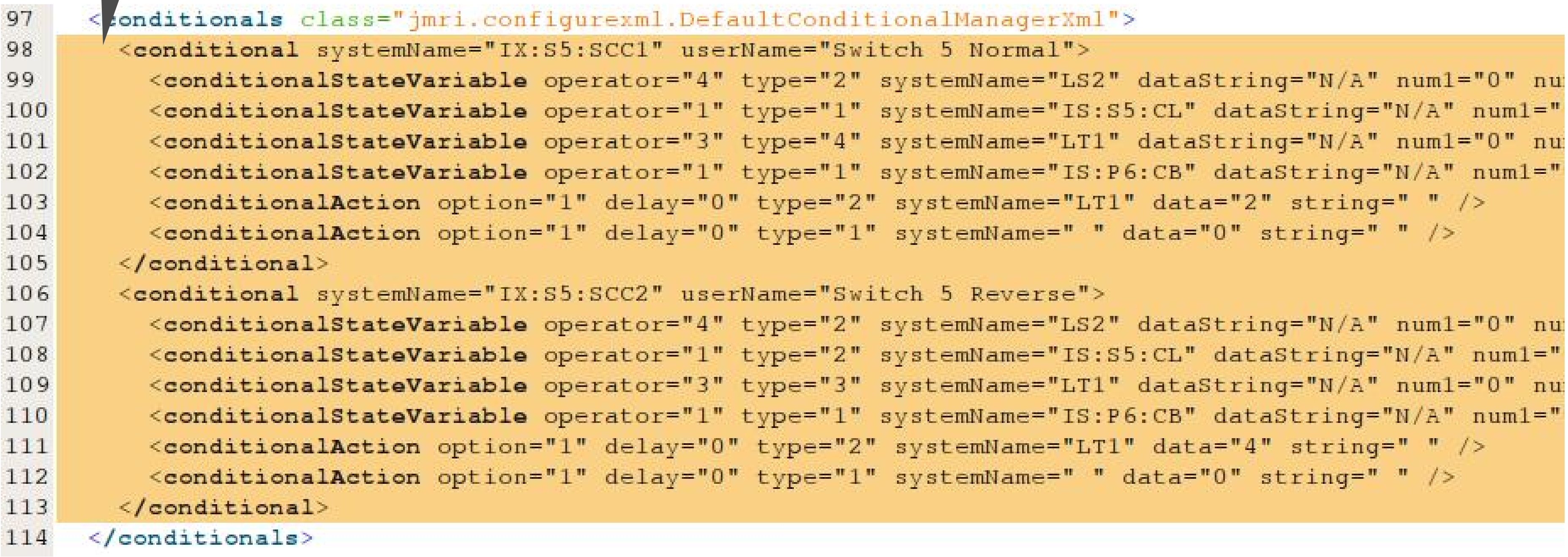
Change each systemName and userName to match our new added positions.   (12 places)

What we have just done is to add 3 new Logix, each with two new conditionals. That means we will also need to add in the 6 new conditionals that they refer to. These new conditionals will be very similar to the first two we created in the last clinic. Each conditional starts with ”<conditional ...” and ends with ”</conditional>”. We will copy the first two conditionals and duplicate them three times.

XML Editing

Manual Editing

Copy and then Paste three times to get three added entries that we can edit to match the new positions.



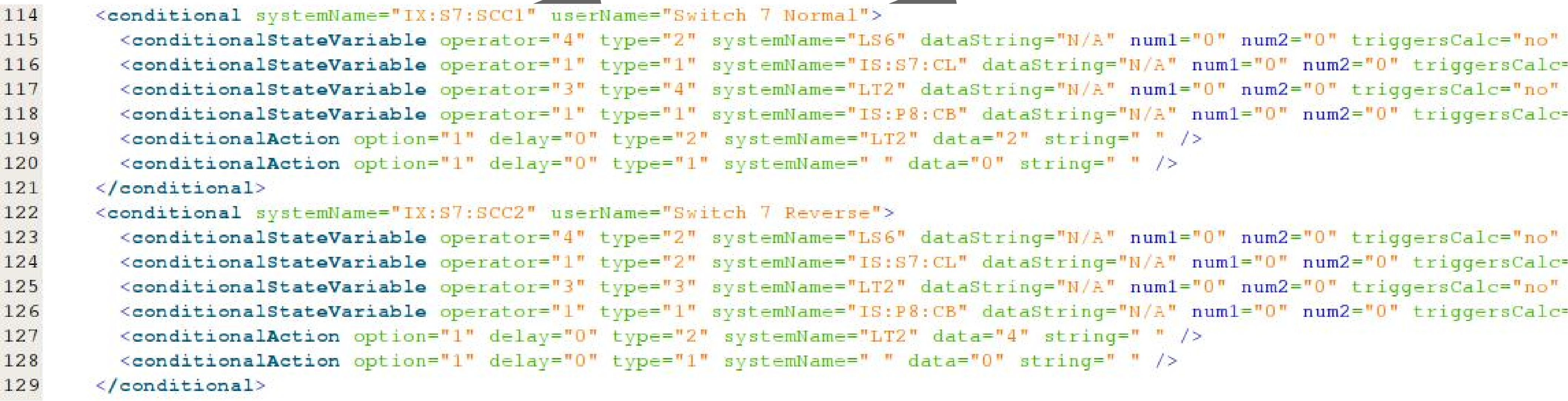
```
<conditionals class="jmri.configurexml.DefaultConditionalManagerXml">
  <conditional systemName="IX:S5:SCC1" userName="Switch 5 Normal">
    <conditionalStateVariable operator="4" type="2" systemName="LS2" dataString="N/A" num1="0" nu
    <conditionalStateVariable operator="1" type="1" systemName="IS:S5:CL" dataString="N/A" num1="
    <conditionalStateVariable operator="3" type="4" systemName="LT1" dataString="N/A" num1="0" nu
    <conditionalStateVariable operator="1" type="1" systemName="IS:P6:CB" dataString="N/A" num1="
    <conditionalAction option="1" delay="0" type="2" systemName="LT1" data="2" string=" " />
    <conditionalAction option="1" delay="0" type="1" systemName=" " data="0" string=" " />
  </conditional>
  <conditional systemName="IX:S5:SCC2" userName="Switch 5 Reverse">
    <conditionalStateVariable operator="4" type="2" systemName="LS2" dataString="N/A" num1="0" nu
    <conditionalStateVariable operator="1" type="2" systemName="IS:S5:CL" dataString="N/A" num1="
    <conditionalStateVariable operator="3" type="3" systemName="LT1" dataString="N/A" num1="0" nu
    <conditionalStateVariable operator="1" type="1" systemName="IS:P6:CB" dataString="N/A" num1="
    <conditionalAction option="1" delay="0" type="2" systemName="LT1" data="4" string=" " />
    <conditionalAction option="1" delay="0" type="1" systemName=" " data="0" string=" " />
  </conditional>
</conditionals>
```

XML Editing

Manual Editing

Copy and then Paste three times to get three added entries that we can edit to match the new positions.

Editing each of the three new pairs of conditionals will be harder than the first time because you will need to change each 'systemName' and 'userName'. (42 places total in our 6 new conditionals.)



```
<conditional systemName="IX:S7:SCC1" userName="Switch 7 Normal">
  <conditionalStateVariable operator="4" type="2" systemName="LS6" dataString="N/A" num1="0" num2="0" triggersCalc="no"
  <conditionalStateVariable operator="1" type="1" systemName="IS:S7:CL" dataString="N/A" num1="0" num2="0" triggersCalc=
  <conditionalStateVariable operator="3" type="4" systemName="LT2" dataString="N/A" num1="0" num2="0" triggersCalc="no"
  <conditionalStateVariable operator="1" type="1" systemName="IS:P8:CB" dataString="N/A" num1="0" num2="0" triggersCalc=
  <conditionalAction option="1" delay="0" type="2" systemName="LT2" data="2" string=" " />
  <conditionalAction option="1" delay="0" type="1" systemName=" " data="0" string=" " />
</conditional>
<conditional systemName="IX:S7:SCC2" userName="Switch 7 Reverse">
  <conditionalStateVariable operator="4" type="2" systemName="LS6" dataString="N/A" num1="0" num2="0" triggersCalc="no"
  <conditionalStateVariable operator="1" type="2" systemName="IS:S7:CL" dataString="N/A" num1="0" num2="0" triggersCalc=
  <conditionalStateVariable operator="3" type="3" systemName="LT2" dataString="N/A" num1="0" num2="0" triggersCalc="no"
  <conditionalStateVariable operator="1" type="1" systemName="IS:P8:CB" dataString="N/A" num1="0" num2="0" triggersCalc=
  <conditionalAction option="1" delay="0" type="2" systemName="LT2" data="4" string=" " />
  <conditionalAction option="1" delay="0" type="1" systemName=" " data="0" string=" " />
</conditional>
```

Manual Editing

This completes the manual editing required to add in the rest of the Logix required for our panel.

XML Editing

Manual Editing

- This completes the manual editing required to add in the rest of the Logix required for our panel.
- Save the file and close your editor.

XML Editing

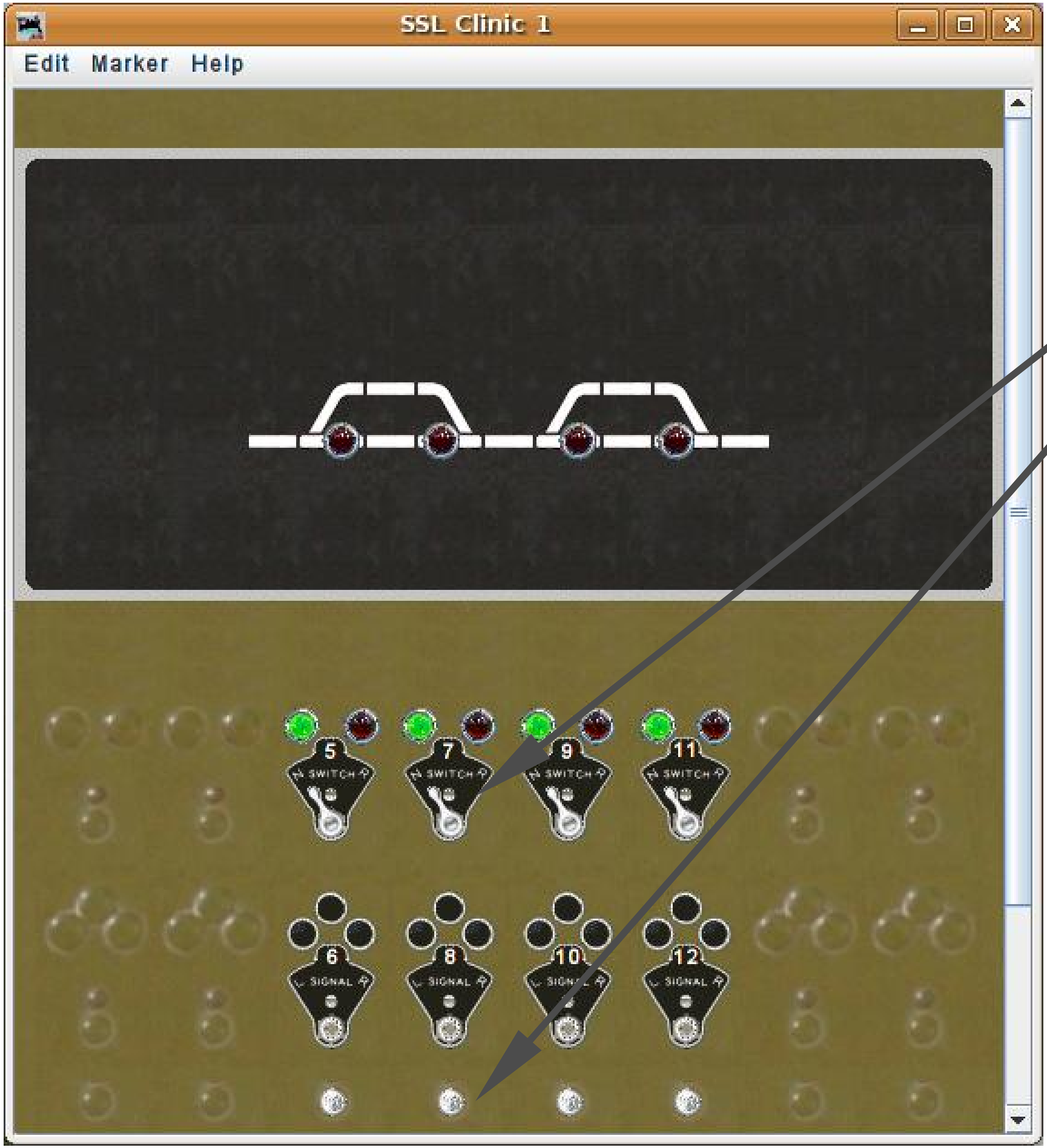
Manual Editing

This completes the manual editing required to add in the rest of the Logix required for our panel.

Save the file and close your editor.

Open PanelPro and load in our new SSL Clinic 1 panel.

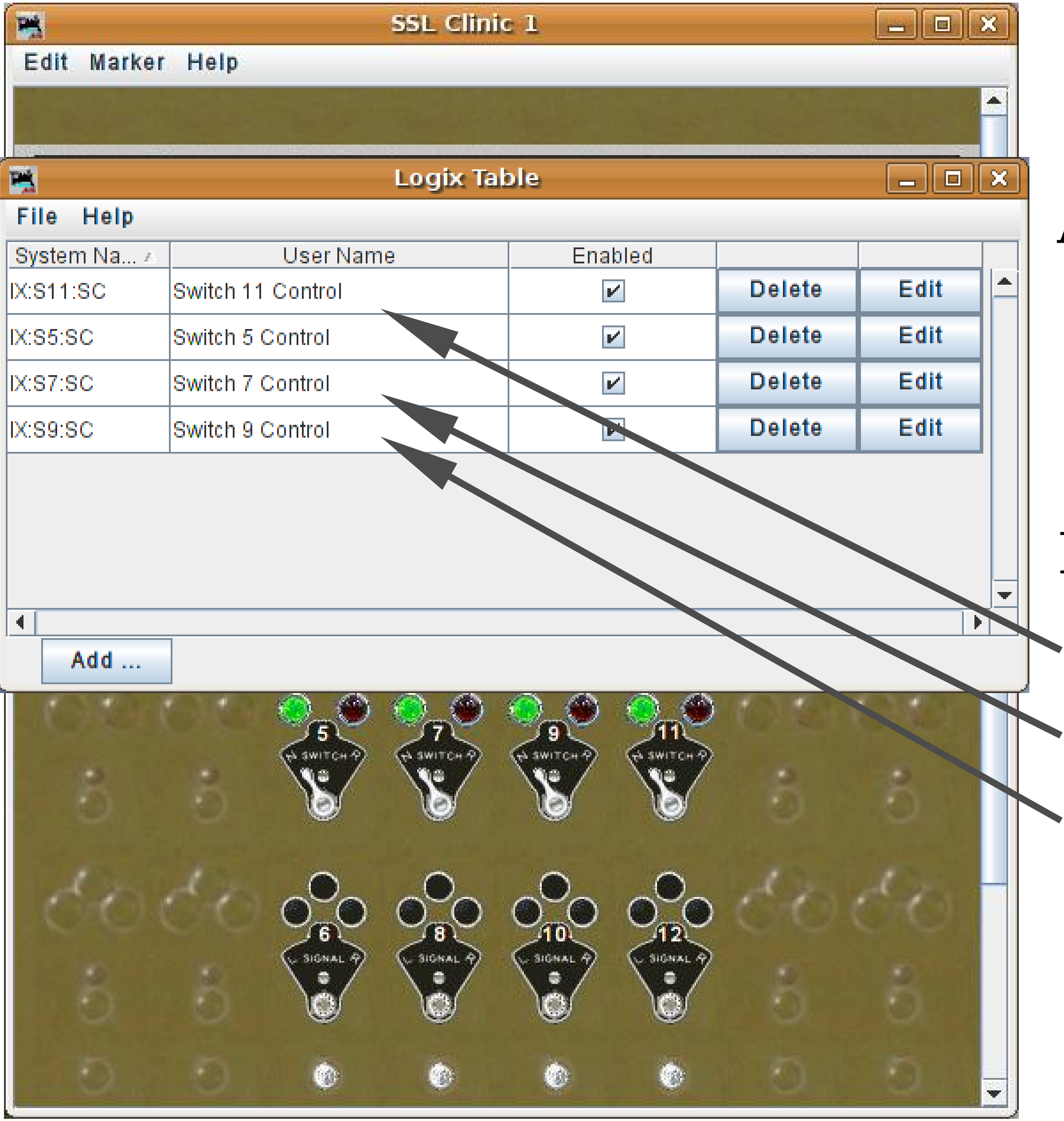
XML Editing



Manual Editing

A bit of testing should show that we now have interlocked control of each turnout, not just the first one.

XML Editing



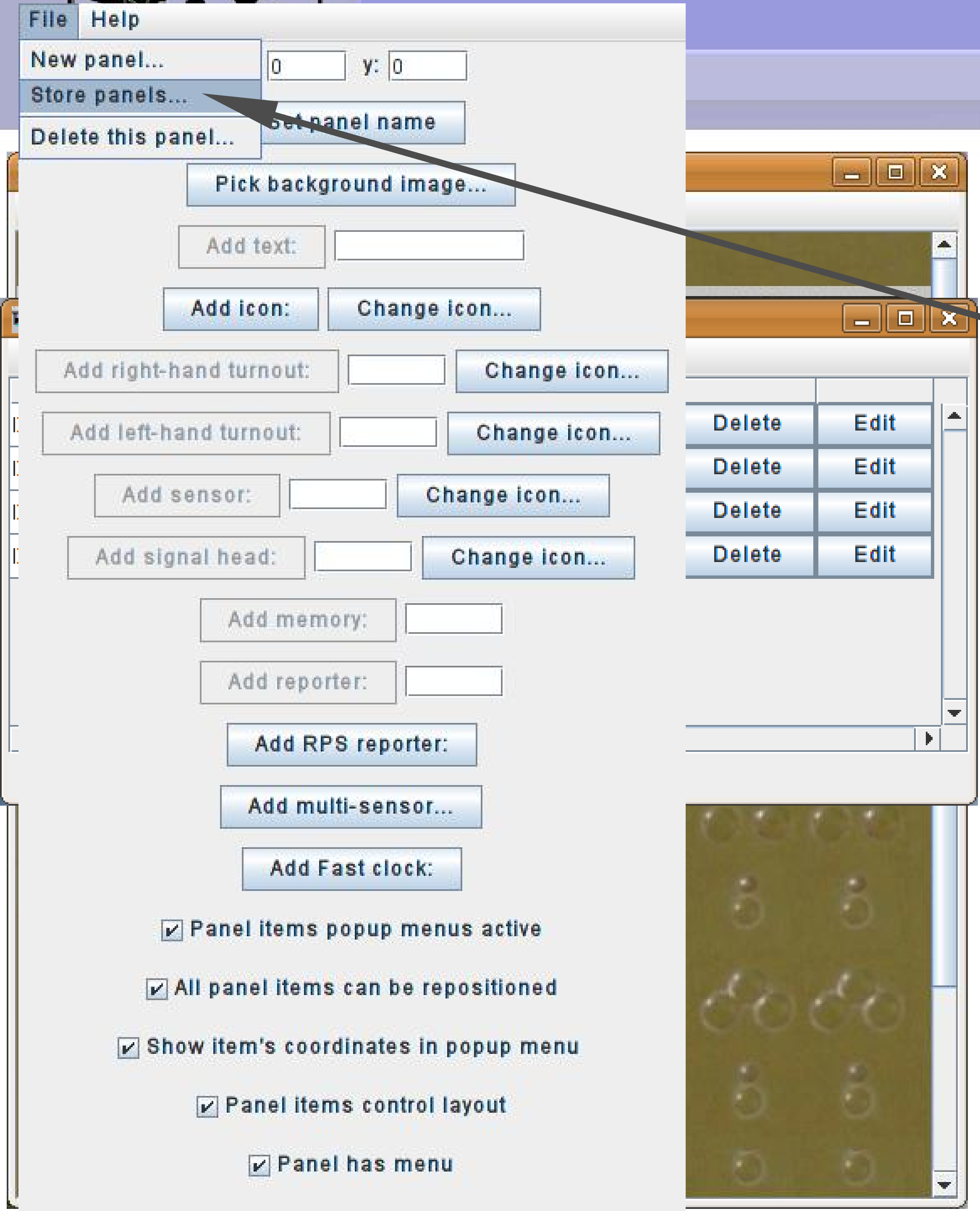
System Na...	User Name	Enabled		
IX:S11:SC	Switch 11 Control	☑	Delete	Edit
IX:S5:SC	Switch 5 Control	☑	Delete	Edit
IX:S7:SC	Switch 7 Control	☑	Delete	Edit
IX:S9:SC	Switch 9 Control	☑	Delete	Edit

Manual Editing

A bit of testing should show that we now have interlocked control of each turnout, not just the first one.

If we open the Logix Table we will see three more entries matching those we manually copied and edited. Note: the sort order of the tables is strictly alpanumeric.

XML Editing



x: 0 y: 0

☑ Panel items popup menus active

☑ All panel items can be repositioned

☑ Show item's coordinates in popup menu

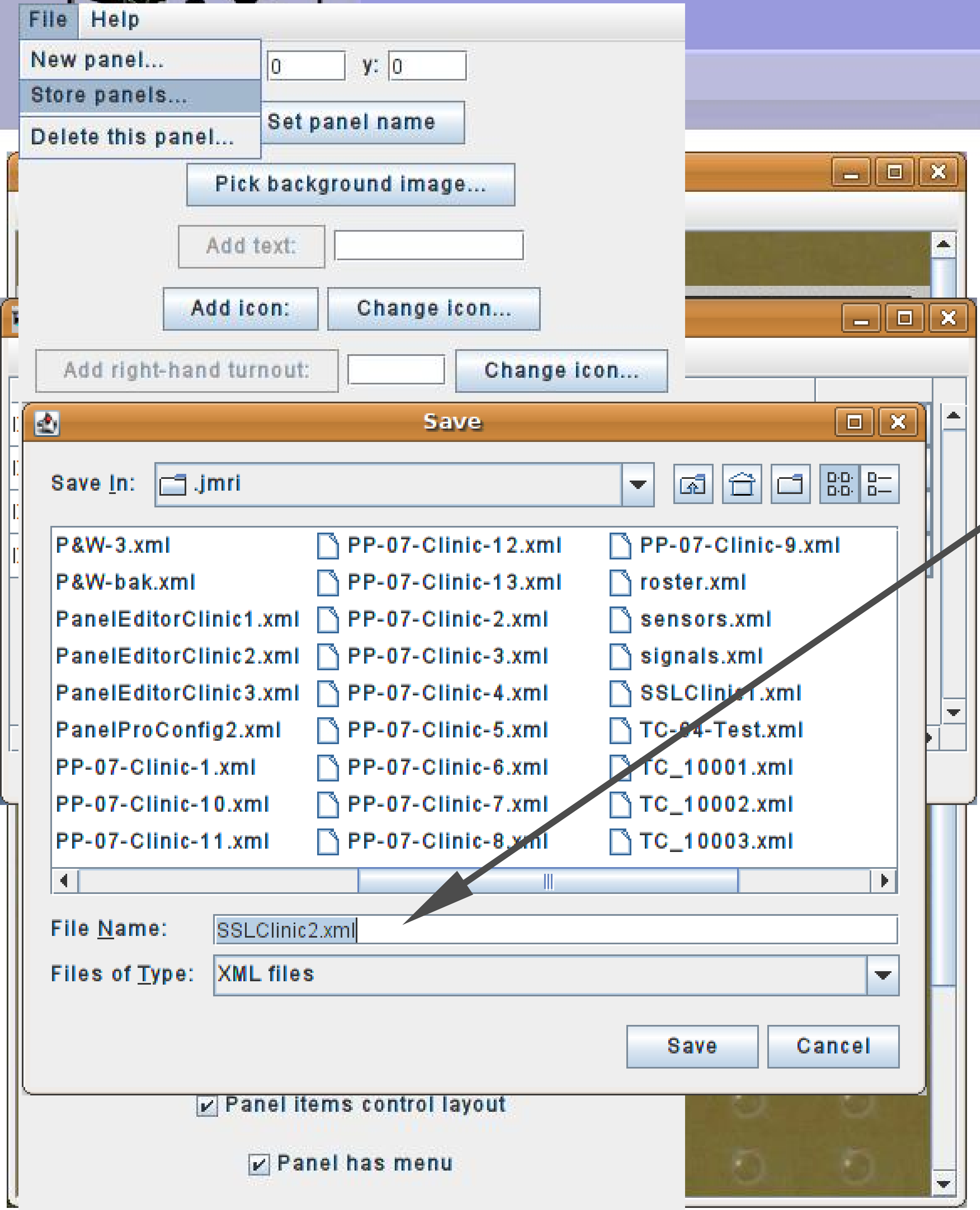
☑ Panel items control layout

☑ Panel has menu

Manual Editing

Save a new copy of the edited panel as SSLClinic2.xml in preparation for the next section of this clinic.

XML Editing



x: 0 y: 0

Save In: .jmri

P&W-3.xml
P&W-bak.xml
PanelEditorClinic1.xml
PanelEditorClinic2.xml
PanelEditorClinic3.xml
PanelProConfig2.xml
PP-07-Clinic-1.xml
PP-07-Clinic-10.xml
PP-07-Clinic-11.xml
PP-07-Clinic-12.xml
PP-07-Clinic-13.xml
PP-07-Clinic-2.xml
PP-07-Clinic-3.xml
PP-07-Clinic-4.xml
PP-07-Clinic-5.xml
PP-07-Clinic-6.xml
PP-07-Clinic-7.xml
PP-07-Clinic-8.xml
PP-07-Clinic-9.xml
roster.xml
sensors.xml
signals.xml
SSLClinic1.xml
TC-04-Test.xml
TC_10001.xml
TC_10002.xml
TC_10003.xml

File Name: SSLClinic2.xml

Files of Type: XML files

☑ Panel items control layout

☑ Panel has menu

Manual Editing

Save a new copy of the edited panel as SSLClinic2.xml in preparation for the next section of this clinic.

XML Editing

- What we have covered so far:
  - Basic manual editing of the panel .xml file.
- Where we are going next:
  - Simple Signal Logic (SSL-clinic-2)