

NaturalONE Tutorial

How to achieve multi-lingual support in NaturalONE Web GUIs?

Recommended Reading

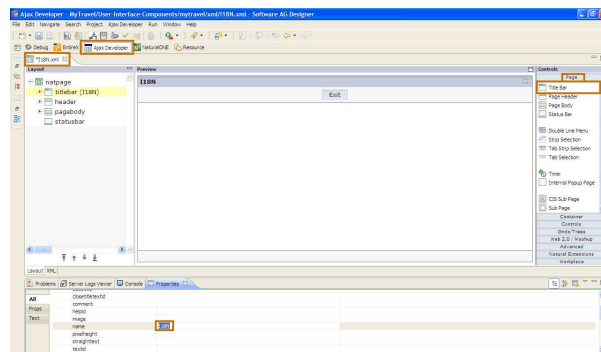
- NaturalONE tutorial: How to set-up a NaturalONE project?
- NaturalONE tutorial: How to create a Web GUI with NaturalONE?
 - Creating a new Layout page -

1 In this tutorial you will learn to create a multi-lingual application. You will start by creating a simple layout consisting of some welcome texts.

Open the **Ajax Developer** perspective.

Create a new page layout called: „**118N**“.

Select the *titlebar* control and modify the *name* property to **118N**.

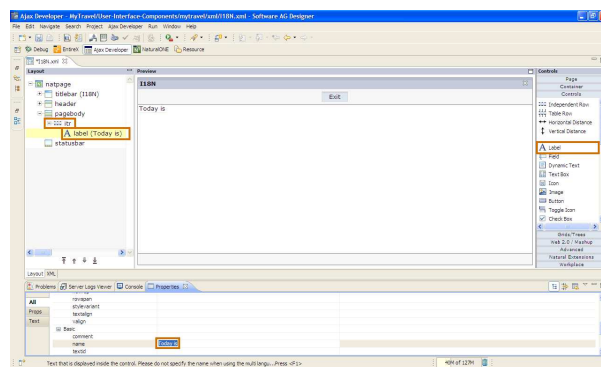


2 Select **Controls** on the **Controls** palette and select the **Label** control icon.

Drag and drop it onto the **Preview** pane.

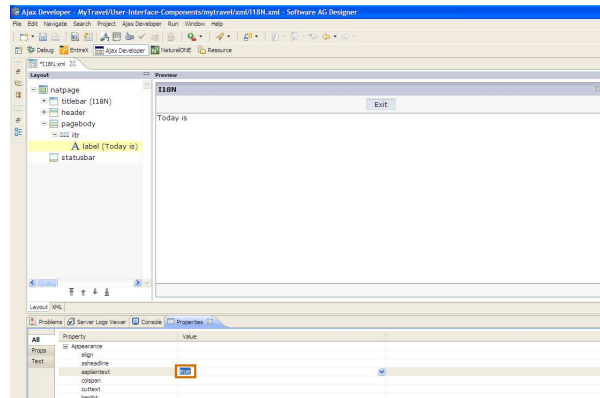
On the **Layout** pane you can see an **ITR** control (which is automatically generated) and a label control as a sub-node.

Select the Label **control** and modify the *name* property: name - “**Today is**”.



- 3 Notice that the text is underlined.

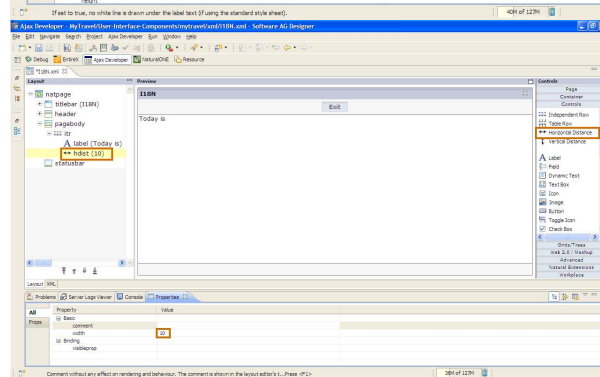
To remove the underline select the *asplaintext* property and set it to **true**.



- 4 Select the **Horizontal Distance** icon, drag and drop it onto the *Preview* area, onto the previously added **Label**.

Then select „Add as a subsequent Node“.

Set the width property of the **hdist** control to **10**.

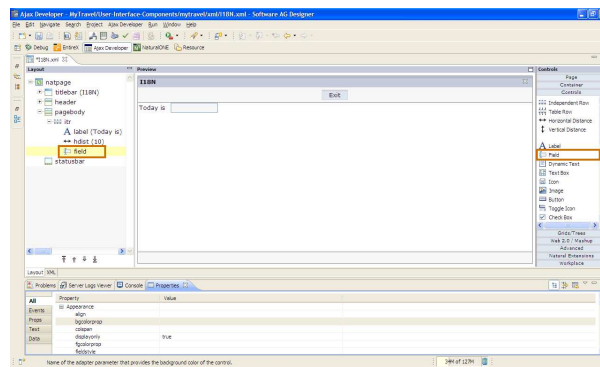


- 5 Perform the same steps for the **field** icon.

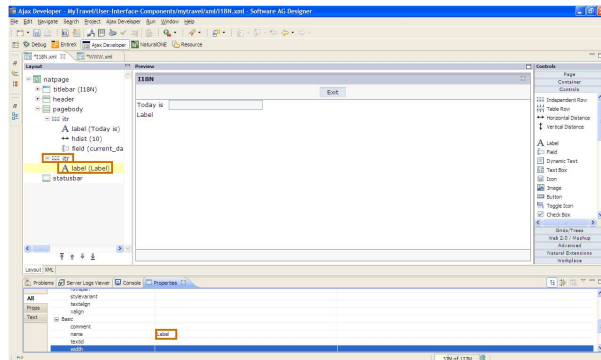
Modify the following properties:

- displayonly - **true**
- fieldstyle - **background-color: white**
- noborder - **true**
- valueprop - **current_date**
- width - **200**

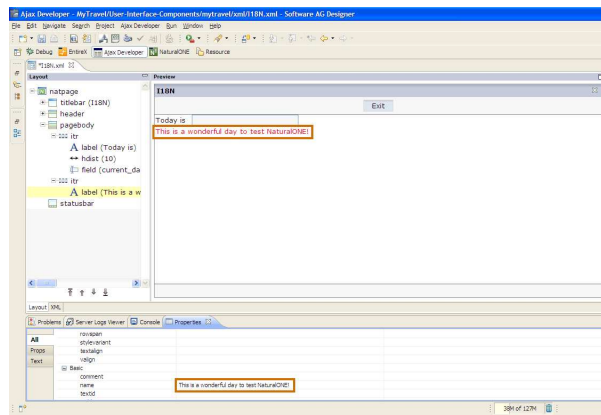
Use the scrollbar on the right-hand side to bring invisible properties into the visible area.



- 6 Perform the same procedure for a new **Label** control, drag it and place it below the previous line: (**field(curren_da...)**).



- 7 Modify the following properties:
- asplintext - **true**
 - labelstyle - **font-size: x-large**
 - color: **red**
 - name - **"This is a wonderful day to test NaturalONE!"**

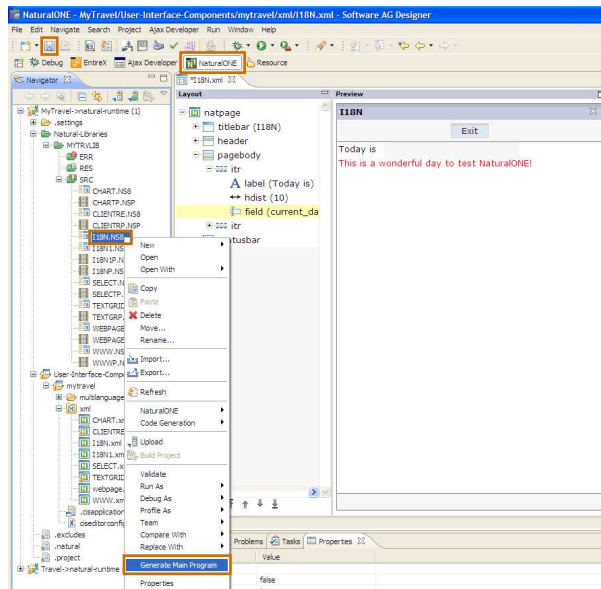


- 8 Now, you can **save** the layout and select back to the **NaturalONE perspective**.

You will create a main program from the generated adapter:

Use the Context menu: **I18N.NS8** → **Generate Main Program**;

Choose the name: **I18NP** and select **OK**.



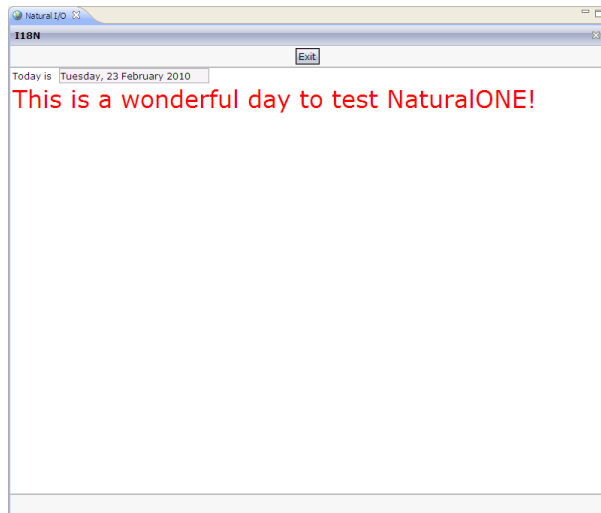
- 9 After END-DEFINE and before the Process Page statement type:

```
*
MOVE EDITED *DATX
(EM=N(9), ' 'ZD' 'L(9)'
'YYYY) TO CURRENT_DATE
```

This will move a displayable date to the layout.

Formatted: English (U.K.)

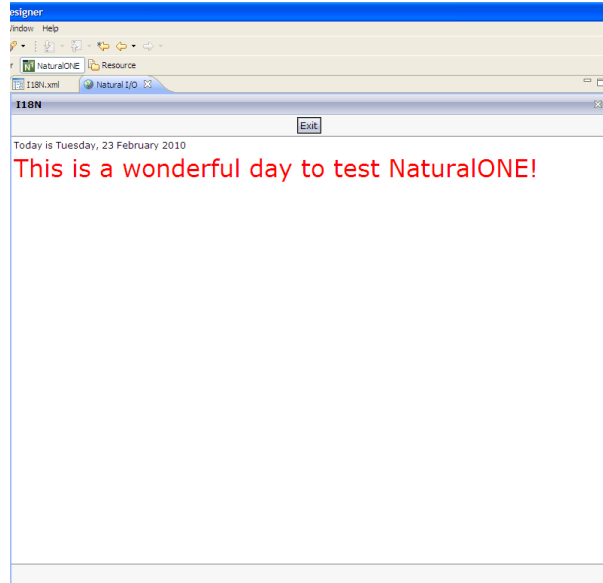
- 10 Save and execute the program.



- 11 You can get rid of the default border and the background color of the date field:

Move back to the layout by choosing **Ajax Developer**, select the **field** control and modify property *fieldstyle* to **background-color: white** and property *noborder* to **true**.

Save the layout, go back to the **NaturalONE perspective** and execute I18NP again.



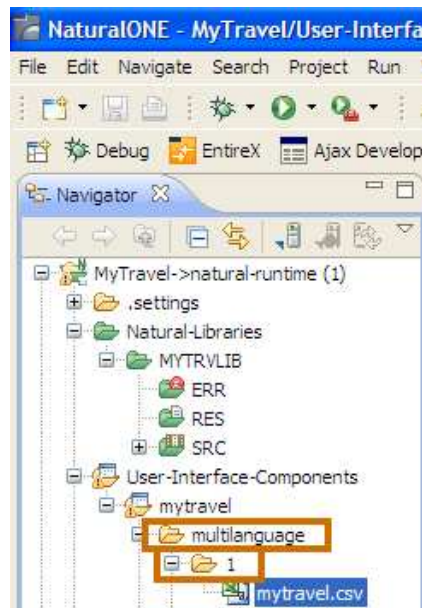
- 12 Now, to the I18N material:
A page contains“
1. “local data”,
text constants defined and handled within the Layout

2. “server side” data,
supplied by the server
(Natural).

The idea is to use *textid* property instead of name which is an explicit label definition within the layout. The Natural for Ajax runtime uses an external translation CSV file which contains the textid and it’s related text string (separated by semicolon).

The lookup is performed taking the Natural *LANGUAGE variable into consideration. This “translation” file is

Formatted: English (U.K.)



placed in a sub-folder named by the respective Natural language code (1 for English, 2 for German, etc.,).

Now, create the necessary folders and CSV files:
When opening multilanguage folder (subfolder of mytravel user interface component), you will see two empty subfolders, en and de - you can remove them and create new subfolders, one for each supported language.

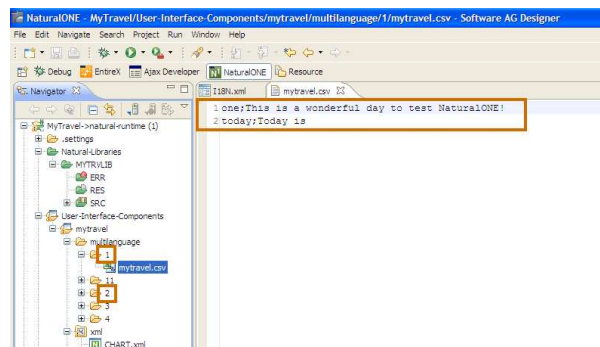
- 13 Select **multilanguage** folder then choose **New → Other → General → Folder**

And set a folder name according to the language (1, 2, ...)

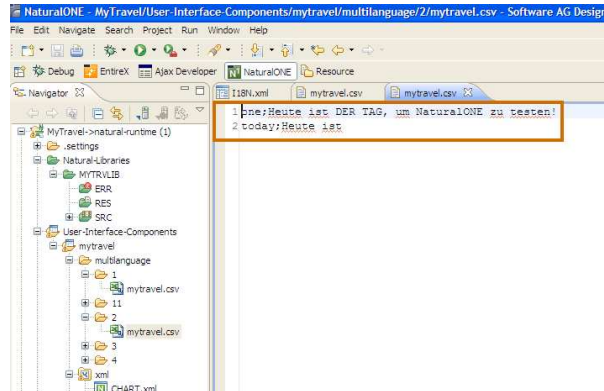
Within each folder create with a text editor (or paste) a translation CSV file containing a row for each textid (here the name of the CSV files is mytravel.csv).

You will use two textid's : one and today and their related labels.

In the image below, you can see the folders structure containing the CSV files for a few languages and one is opened displaying the content.



- 14 Here you will edit the German version (*LANGUAGE=2):



- 15 Following are the texts for different languages;

For English(1):

one;This is a wonderful day
to test NaturalONE!
today;Today is

For German(2):

one;Heute ist DER TAG, um
NaturalONE zu testen!
today;Heute ist

For French(3):

one;Quelle bonne journée
pour tester NaturalOne!
today;Aujourd'hui c'est le

For Spanish(4):

one;Hoy es un día
maravilloso para probar
NaturalONE!
today;Hoy es

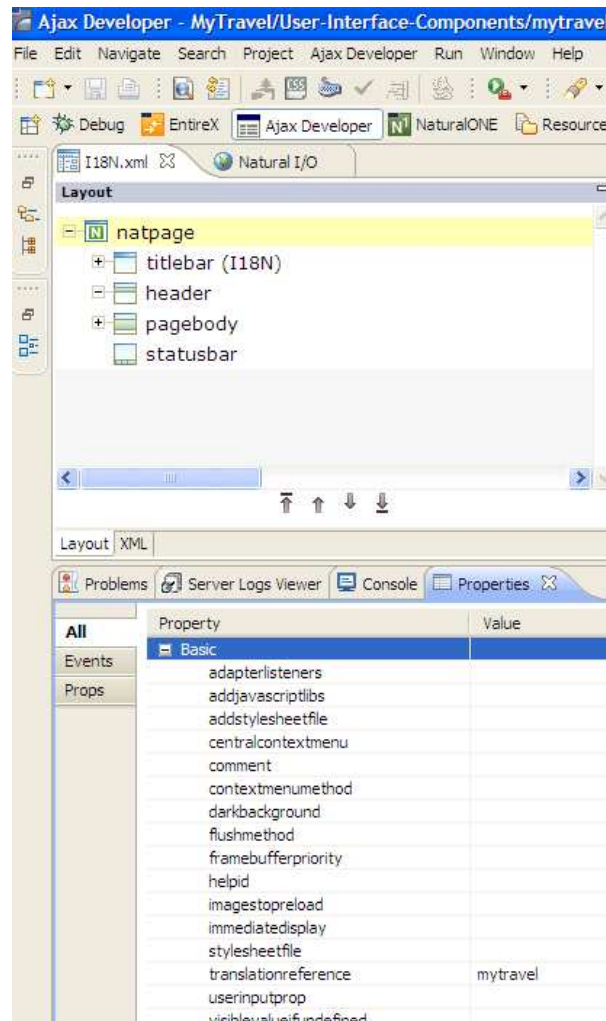
For Portuguese(11):

one;Este é um dia
maravilhoso para testar o
NaturalONE!
today;Hoje é

- 16 Now, modify the layout in order to use this I18N method:

Re-open **I18N.xml** (if closed) and select the **natpage** node

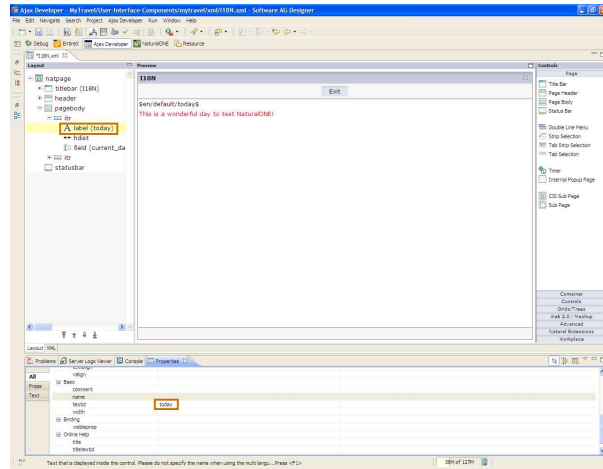
Modify the *translationreference* property to: **mytravel** (the translation filename)



17 Now, select the “**Today**” label.

Erase the *name* property.

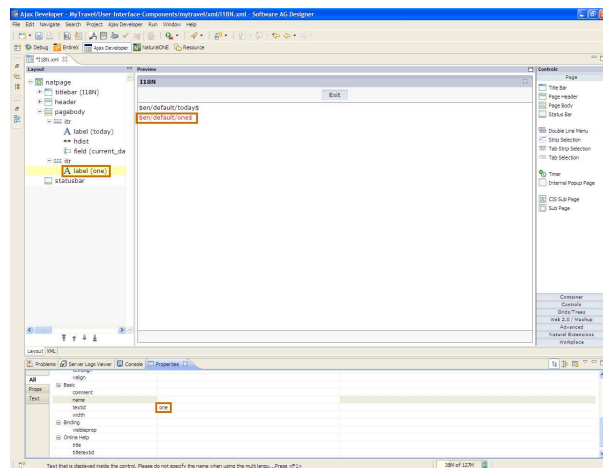
Set *textid* property to **today**.



18 Do the same with the “**This is...**” label

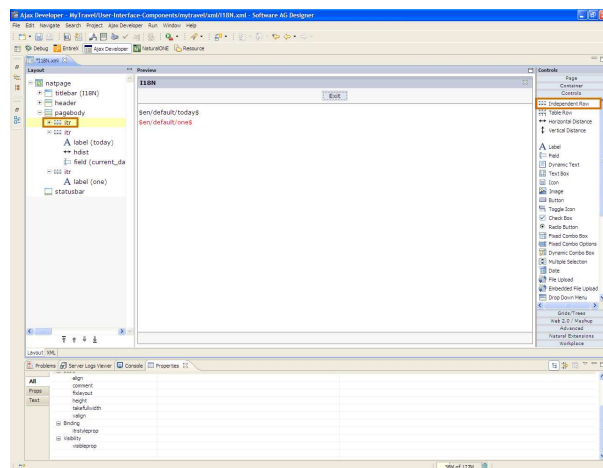
Erase the *name* property.

Set the *textid* property to **one**.

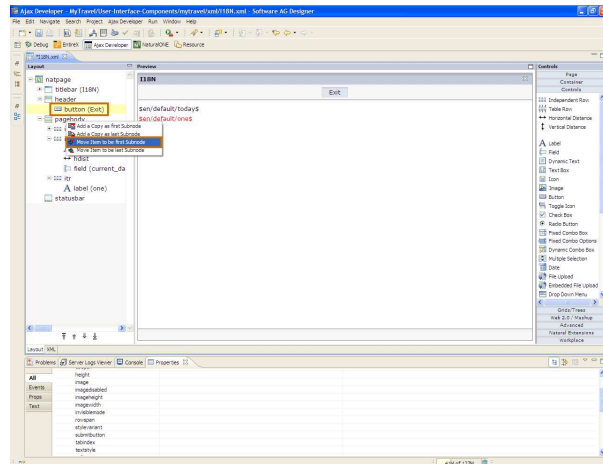


19 In order to enable a dynamic setting of the *LANGUAGE variable:

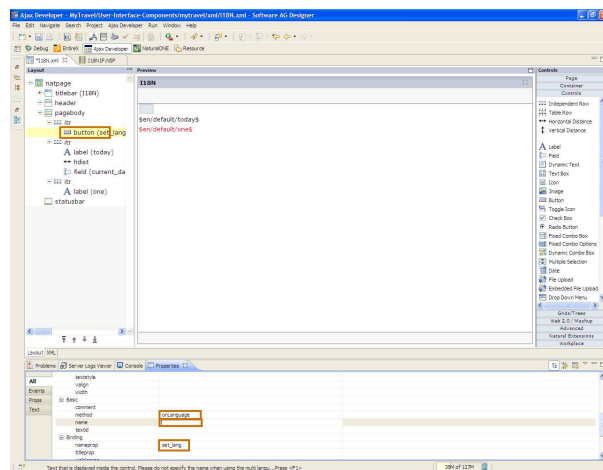
Create a new *itr* node (drag & drop onto pagebody node).



- 20 Drag and drop button: **Exit** control onto the new itr and select **Move Item to be first Subnode**.



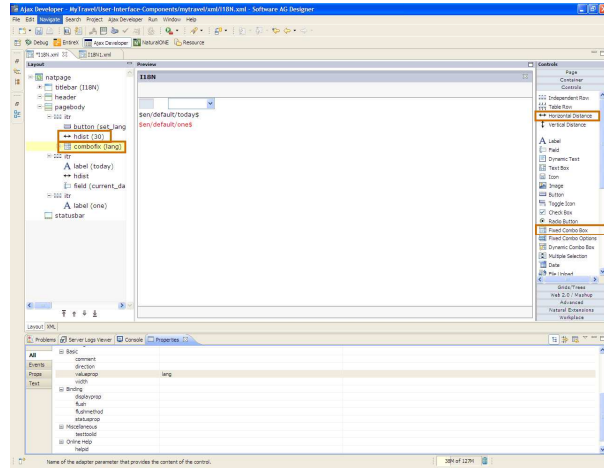
- 21 Erase the button's label from name property; you could use the textid property as before, but instead use nameprop to demonstrate server's (Natural) part in I18N process - this property will contain the variable name (set_lang) in which the Natural application will do the translation and set an appropriate text. You will also change the method name to onLanguage.



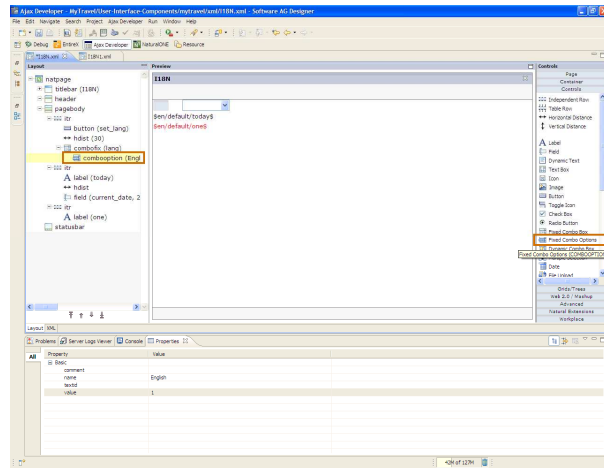
- 22 Now add some horizontal distance using the **hdist** control and a **Fixed Combo Box** (combobox).

Select both from Controls Palette → Controls.

For the combobox control, you will use only the valueprop property which is the name of the variable(lang) in which the selected value is returned.

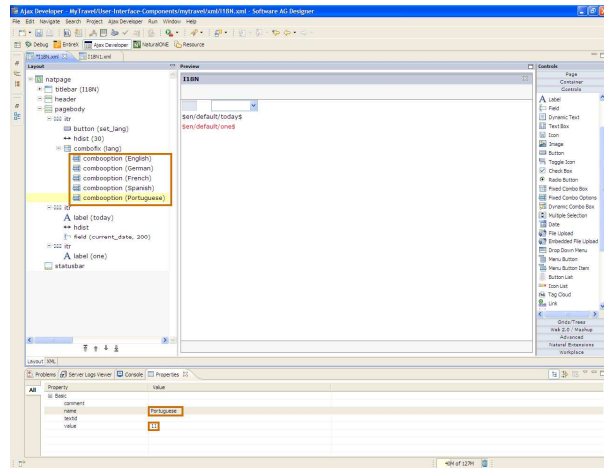


- 23 Now drag and drop a **Fixed Combo Options** icon onto the combobox node; combobox control has two properties of interest: name, containing the displayed label and value which is returned after selection - .



- 24 You will use as many as needed, 5 in our case.

At the end, **save** the modified layout and select back the **NaturalONE** perspective.



- 25 You can manually modify your previous I18NP by adding the new fields you added to the layout but, in this case, it is simpler to generate again (step 8).

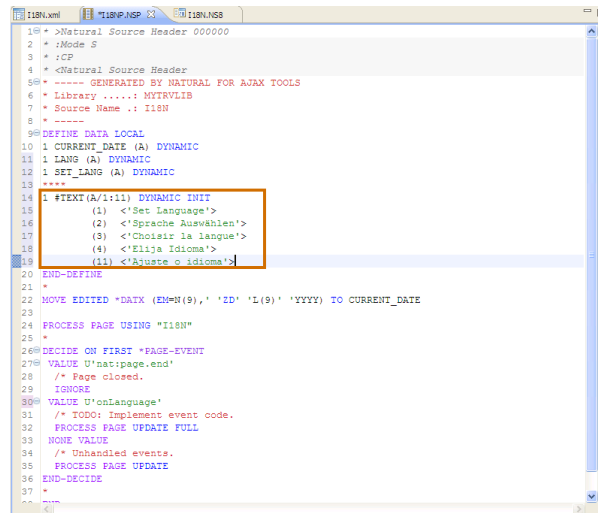
Now to the programs logic: In order to provide the language-dependent strings, define an array for each supported language value, that will contain the correct text.

The array should be:

```

****
1 #TEXT(A/1:11) DYNAMIC
INIT
  (1)      <'Set
           Language'>
  (2)      <'Sprache
           Auswählen'>
  (3)      <'Choisir la
           langue'>
  (4) <'Elija Idioma'>
  (11) <'Ajuste o idioma'>

```



- 26 Create a small subroutine to edit the date and set the button's label to the relevant array entry.

```

DEFINE SUBROUTINE SET-
LANGUAGE
MOVE EDITED *DATX
(EM=N(9), ' 'ZD' 'L(9)'
'YYYY) TO CURRENT_DATE
#LANG := *LANGUAGE
SET_LANG := #TEXT(#LANG)
END-SUBROUTINE
    
```

```

90 DEFINE DATA LOCAL
10 1 CURRENT_DATE (A) DYNAMIC
11 1 LANG (A) DYNAMIC
12 1 SET_LANG (A) DYNAMIC
13 *****
14 1 #LANG(11)
15 1 #TEXT(A/1:11) DYNAMIC INIT
16 (1) <'Set Language'>
17 (2) <'Sprache Auswählen'>
18 (3) <'Choisir la langue'>
19 (4) <'Elija Idioma'>
20 (11) <'Ajuste o idioma'>
21 END-DEFINE
22 *
23 PERFORM SET-LANGUAGE
24 *
25 PROCESS PAGE USING "i18n"
26 *
27 @ DECIDE ON FIRST *PAGE-EVENT
28 @ VALUE U'nat:page.end'
29 /* Page closed.
30 IGNORE
31 @ VALUE U'onLanguage'
32 /* TODO: Implement event code.
33 PROCESS PAGE UPDATE FULL
34 NONE VALUE
35 /* Unhandled events.
36 PROCESS PAGE UPDATE
37 END-DECIDE
38 DEFINE SUBROUTINE SET-LANGUAGE
39 MOVE EDITED *DATX (EM=N(9), 'ZD' 'L(9)' 'YYYY) TO CURRENT_DATE
40 #LANG := *LANGUAGE
41 SET_LANG := #TEXT(#LANG)
42 END-SUBROUTINE
43 *
44 END
    
```

- 27 For the event 'onLanguage', sent when pressing the button, Set *LANGUAGE to the value provided by the cobobox.

```

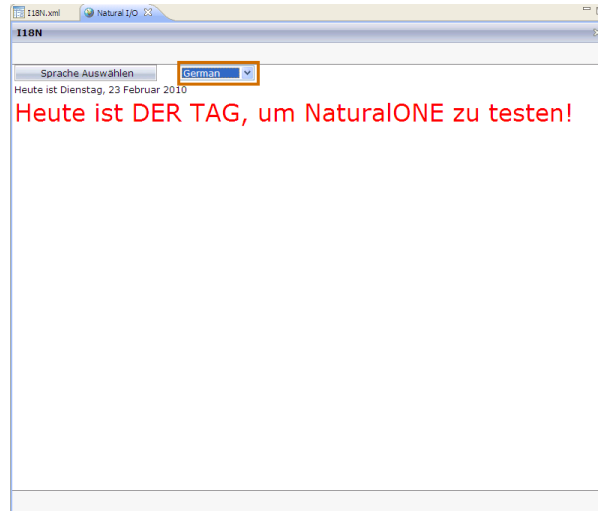
*LANGUAGE := VAL(LANG)
PERFORM SET-LANGUAGE
    
```

Save the modified program and **execute**.

```

90 DEFINE DATA LOCAL
10 1 CURRENT_DATE (A) DYNAMIC
11 1 LANG (A) DYNAMIC
12 1 SET_LANG (A) DYNAMIC
13 *****
14 1 #LANG(11)
15 1 #TEXT(A/1:11) DYNAMIC INIT
16 (1) <'Set Language'>
17 (2) <'Sprache Auswählen'>
18 (3) <'Choisir la langue'>
19 (4) <'Elija Idioma'>
20 (11) <'Ajuste o idioma'>
21 END-DEFINE
22 *
23 PERFORM SET-LANGUAGE
24 *
25 PROCESS PAGE USING "i18n"
26 *
27 @ DECIDE ON FIRST *PAGE-EVENT
28 @ VALUE U'nat:page.end'
29 /* Page closed.
30 IGNORE
31 @ VALUE U'onLanguage'
32 *LANGUAGE := VAL(LANG)
33 PERFORM SET-LANGUAGE
34 PROCESS PAGE UPDATE FULL
35 NONE VALUE
36 /* Unhandled events.
37 PROCESS PAGE UPDATE
38 END-DECIDE
39 DEFINE SUBROUTINE SET-LANGUAGE
40 MOVE EDITED *DATX (EM=N(9), 'ZD' 'L(9)' 'YYYY) TO CURRENT_DATE
41 #LANG := *LANGUAGE
42 SET_LANG := #TEXT(#LANG)
43 END-SUBROUTINE
44 *
45 END
    
```

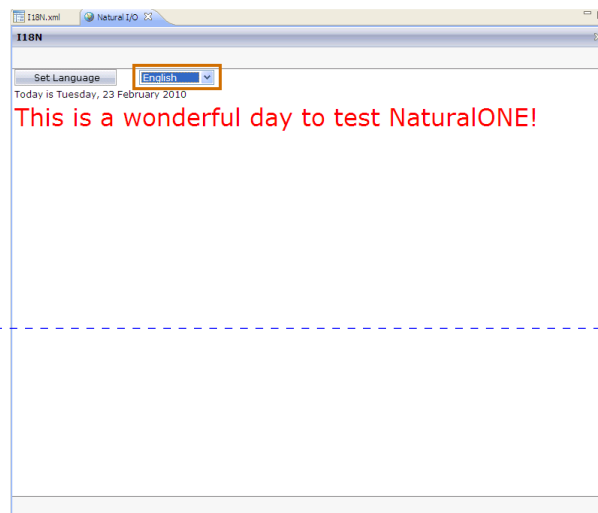
- 28 Select **German** from the Combobox and press the button.



- 29 Now, check for English or any other defined language.

Observe that the button label (provided by the server) and the labels, based upon textids, are correctly translated.

Also observe that the current date provided by Natural, is also language sensitive and correctly translated (automatically).



Formatted: English (U.K.)