

Multilingual XForms



Contents

1. Contents
2. Technique
3. Translate
4. An Example
5. Prepare
6. Itemset
7. Translate
8. Switch languages
9. Result
10. Initialising

Technique

Everywhere you have a `<label>` element with text, you can replace it with a `<label>` with a `ref` attribute, or using an output. Instead of:

```
<label>How can you help?</label>
```

you put the text in an instance:

```
<instance id="m">
  <messages xmlns="" lang="en">
    ...
    <help>How can you help?</help>
    ...
  </messages>
</instance>
```

and use an output in the label:

```
<label><output ref="instance('m')/help"/></label>
```

or a ref:

```
<label ref="instance('m')/help"/>
```

Translate

Then you translate the messages instance into as many languages as you like, and store each in a file.

To switch language, you use a submission to replace the messages instance with one from one of the files:

```
<submission replace="instance" instance="m" resource="french.xml"  
method="get" serialization="none"/>
```

The `serialization="none"` is so that no data from the default instance gets sent with the submission

`method="get"` is the default, so is not strictly needed.

An Example

Say you have an XForm that starts like this:

```
Please help us discover problems and solutions that would improve our proces

<select1 ref="choice" appearance="full">
  <label>How can you help?</label>
  <item><label>You know a problem that needs to be fixed</label>
    <value>problem</value></item>
  <item><label>You know a 'solution' that doesn't work</label>
    <value>failure</value></item>
  <item><label>You have a prediction about a future possible failure</label>
    <value>prediction</value></item>
</select1>
```

Which looks like this:

Please help us discover problems and solutions to improve our processes.

How can you help?

- ☐ You know a problem that needs to be fixed
- ☐ You know a 'solution' that doesn't work
- ☐ You have a prediction about a future possible failure

Submit

[Source](#)

Prepare

You create an instance to contain your current messages:

```
<instance id="m">
  <messages xmlns="" lang="en">
    <intro>Please help us discover problems and solutions to improve our pro
    ...
  </messages>
</instance>
```

Now you can change the first line to this:

```
<output ref="instance('m')/intro"/>

<select1 ref="choice" appearance="full">
  <label>How can you help?</label>
  <item><label>You know a problem that needs to be fixed</label>
    <value>problem</value></item>
  <item><label>You know a 'solution' that doesn't work</label>
    <value>failure</value></item>
  <item><label>You have a prediction about a future possible failure</label>
    <value>prediction</value></item>
</select1>
```

Itemset

For the select1, we add a number of messages to the message instance, one for the label, and one each for the items, along with the value that it matches:

```
<instance id="m">
  <messages xmlns="" lang="en">
    <intro>Please help us discover problems and solutions to improve our p
    <choice>
      <label>How can you help?</label>
      <item value="problem">You know a problem that needs to be fixed</it
      <item value="failure">You know a 'solution' that doesn't work</item
      <item value="prediction">You have a prediction about a future possi
    </choice>
  </messages>
</instance>
```

Now you can use a ref for the label, and an itemset to replace the items in the select1:

```
<select1 ref="choice">
  <label ref="instance('m')/choice/label"/>
  <itemset ref="instance('m')/choice/item">
    <label ref="."/>
    <value ref="@value"/>
  </itemset>
</select1>
```

Translate

Then you translate the messages instance into as many languages as you wish, store them in files, and record the languages you have used:

```
<languages>
  <language file="messages-ca.xml">Català</language>
  <language file="messages-de.xml">Deutsch</language>
  <language file="messages-en.xml">English</language>
  <language file="messages-es.xml">Español</language>
  <language file="messages-fr.xml">Français</language>
  <language file="messages-it.xml">Italiano</language>
  <language file="messages-hu.xml">Magyarul</language>
  <language file="messages-nl.xml">Nederlands</language>
  <language file="messages-zh.xml">简体中文</language>
</languages>
```

Store the above instance in a file, and load it:

```
<instance id="languages" src="languages.xml"/>
```

Switch languages

Create a control to switch language:

```
<select1 ref="file">
  <label>?</label>
  <itemset ref="instance('languages')/language">
    <label ref="."/>
    <value ref="@file"/>
  </itemset>
  <action ev:event="xforms-value-changed">
    <send submission="change-language"/>
  </action>
</select1>
```

Whenever this control gets used, and a language is selected, the instance value `file` gets changed, and the send action causes the following submission to be initiated, which replaces the messages instance with the translated version in the selected file:

```
<submission id="change-language" replace="instance" instance="m"
  method="get" serialization="none">
  <resource value="file"/>
</submission>
```

Result

Here it is in use:

流程改进

请帮助我们发现问题，并找到能改善我们的流程的解决方案  简体中文 ▼
如何才能帮助我们呢？

- ☐ 了解需要被修正的问题
- ☐ 发现某个“解决方案”并不能解决问题
- ☐ 做出关于未来失败可能性的预测

[Source](#)

Initialising

Note that if someone changes language and then wants to change back again, you have to supply a `messages-en.xml` file as well, and so you may as well initialise the messages instance with that:

```
<instance id="m" src="messages-en.xml"/>
```

Whenever you need to add another language:

- translate the messages instance
- store it in a new file
- add the language to the languages file.

Easy! The XForm then works without change with the new language.