



Les "processing instructions"

1. Exemple instruction If

Cette image montre comment poser une [instruction If](#). La condition est définie dans l'attribut "Condition" de l'élément "If".

The screenshot displays the configuration of an 'If' instruction in a software interface. The left pane shows a tree view of the instruction's content, and the right pane shows the 'Properties' window with the 'Condition' attribute set to ' ((\$tokens_list_size)>0) && (\$getTokens_txt2)!= null)'. The tree view shows an 'If' instruction with a condition, branching into 'If true' and 'If false'. The 'If true' branch contains a 'Loop' with a 'QuestionSet' containing a 'Question' with a 'Label' containing a 'Token' and a 'Text' box with 'token[\$t]', and a 'Text' box with 't = \$t+1'. The 'If false' branch contains a 'Table' with 'Column Definitions' and a 'TableBody' with a 'Row' containing a 'Td' with a 'Paragraph' containing 'Désolé!' and 'aucune liste à retourner.'

2. Exemple instruction IfExist

Afficher un paragraphe selon l'existence d'une variable.

L'expression de la variable prénom est:

La variable prénom n'existe pas

The screenshot displays an XML editor interface. The main workspace shows a tree view for an `IfExist $($prenom)` instruction. It is divided into two branches: `If true` and `If false`. The `If true` branch contains a `Table` element with a `Column Definition` and a `TableBody` containing a `Row` with a `Td` containing a `Paragraph` with the text "L'expression de la variable prénom est:" and a variable reference `$(prenom)`. The `If false` branch contains a `Table` element with a `TableBody` containing a `Row` with a `Td` containing a `Paragraph` with the text "La variable prenom n'existe pas". To the right, the `Properties` window shows the `IfExist` instruction with the `Condition` property set to `$(prenom)`. The `Project Explorer` on the far right lists various XML files, including `ReferenceForm.xml`.

3. Exemple instruction Loop

Exemple de boucle Loop.

La variable $\$(t)$ est de type entier et initialisée à 0 pour servir d'itérateur; la variable $\$(tokens_list_size)$ est également de type entier et représente la taille d'une liste. L'instruction Loop permet de lister tous les items de la liste.

The screenshot shows a software interface for configuring a Loop instruction. The main window displays a flowchart with an 'If true' block containing a 'Loop' block. The 'Loop' block contains a 'QuestionSet' block with a 'Question' block, which includes a 'Label' block with a 'Token' button, a '\$(t)' variable, and a 'Text box: token[\$(t)]'. Below the 'QuestionSet' is a 't = \$(t) + 1' block. The 'Properties' panel on the right shows the 'Loop' instruction with the condition '\$(t) < \$(tokens_list_size)'. A red line connects the 'Loop' block in the flowchart to the 'Loop' instruction in the properties panel.

4. Exemple instruction Foreach

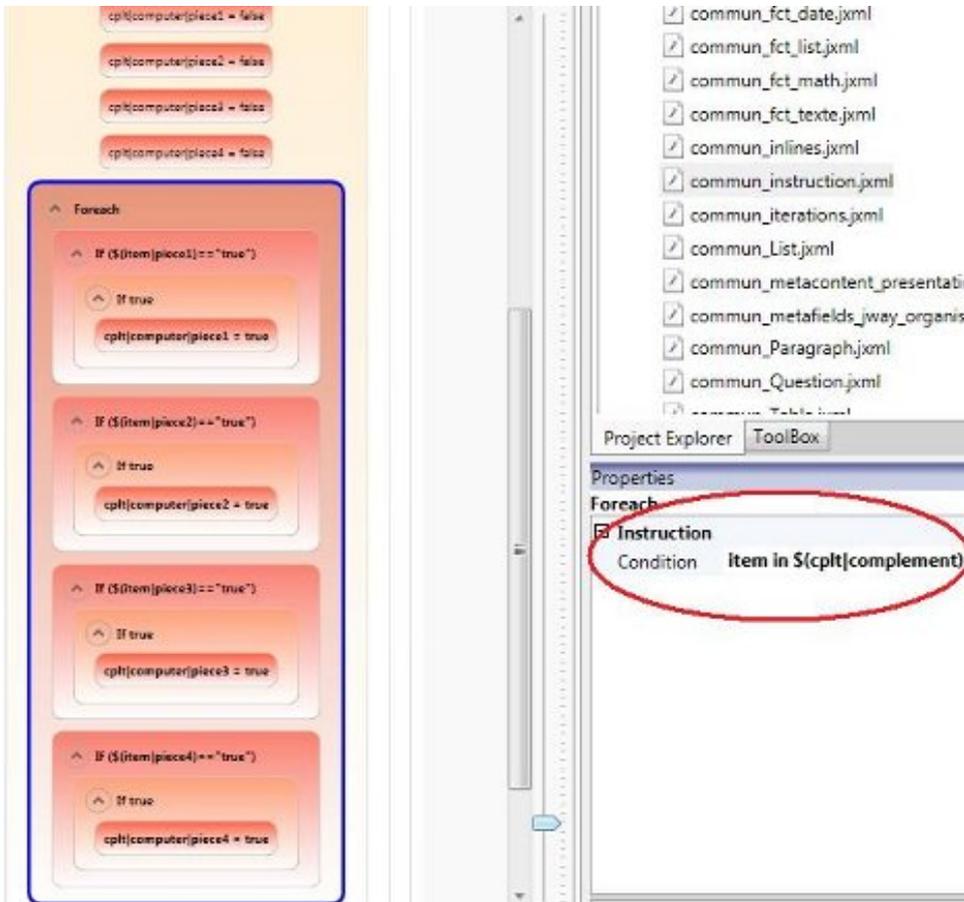
Répétition d'un bloc d'éléments tant qu'un item est dans la liste.

Cet exemple illustre l'utilisation de l'instruction Foreach. Nous avons une liste de pièces à sélectionner. Pour permettre plusieurs sélections, nous avons utilisé un Table itératif. Par contre, une même pièce ne peut être sélectionnée plus d'une fois.

Date réception	Liste des pièces jointes																				
<table border="1"> <tr><td> </td><td>/</td><td> </td><td>/</td><td> </td><td> </td></tr> </table>		/		/																	<input type="checkbox"/> Demande <input type="checkbox"/> Extrait d'acte de naissance <input type="checkbox"/> Certificat de résidence <input type="checkbox"/> Photo d'identité
	/		/																		
<table border="1"> <tr><td> </td><td>/</td><td> </td><td>/</td><td> </td><td> </td></tr> </table>		/		/																	<input type="checkbox"/> Demande <input type="checkbox"/> Extrait d'acte de naissance <input type="checkbox"/> Certificat de résidence <input type="checkbox"/> Photo d'identité
	/		/																		

Les images ci-dessous montrent comment l'exemple a été implémenté.

L'instruction Foreach:



Le Table itératif:

The screenshot displays a software interface for editing a data table. The table has a header row and three data rows. Each data row contains a 'Question' field and a 'Suffix' field. The 'DataIterator' component is highlighted in red, and a red line points to its properties in the 'Properties' window on the right. The 'Properties' window shows the 'DataPath' property set to 'cplt|complement'.

Question	Suffix
CheckBox et piece1	Demande
CheckBox et piece2	Extrait acte de naissance
CheckBox et piece3	

Project Explorer

- commun_inlines.xml
- commun_instruction.xml
- commun_iterations.xml
- commun_List.xml
- commun_metacontent_present
- commun_metafields_iway_orga
- commun_Paragraph.xml
- commun_Question.xml
- commun_Table.xml
- commun_test.xml
- Document_configuration_imag
- glossaire.properties
- ReferenceForm.xml

Properties

DataIterator

General

DataPath **cplt|complement**

InternalName **elt**

MaxItemAllowe **12**

MinItemToDisp **1**

MinItemToPrint **2**