10 March 2016
Author: JC Dauphin
jc.dauphin@gmail.com

J-ISIS 10 March 2016

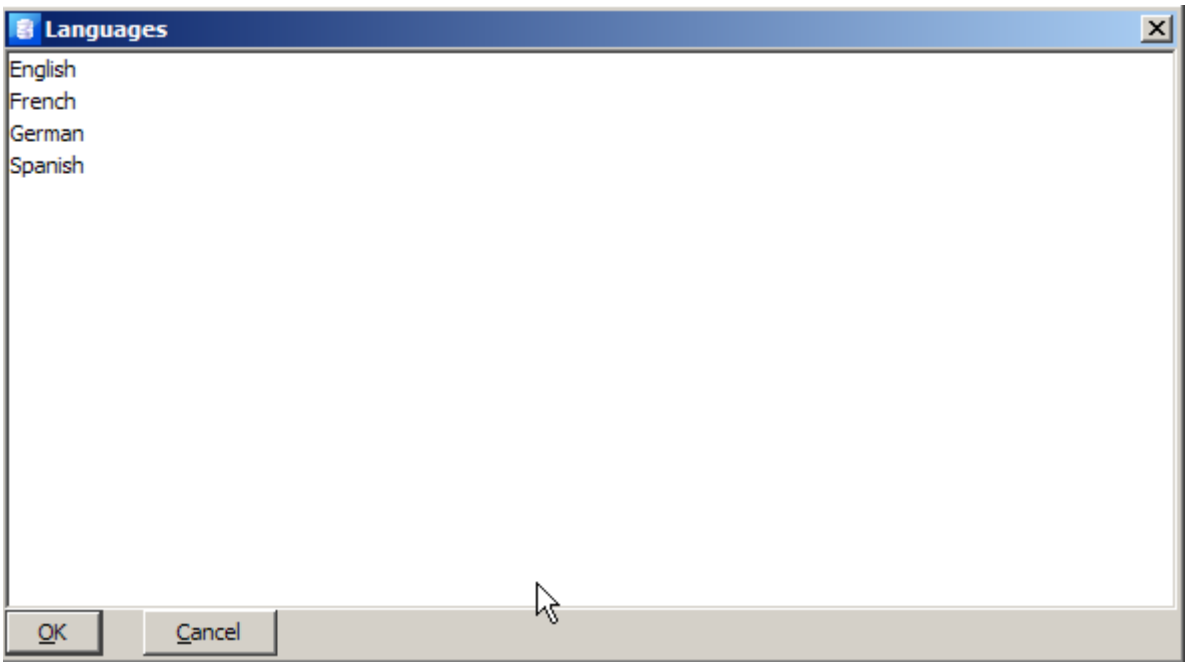
17. Pick Lists for Data Entry and Data Validation

17.1 Introduction

Pick-list definition. A pick-list enables the user to display a list giving a choice of values for a particular field or subfield during data entry. In the pick-list definition box you enter **choice** followed by *two colons* and a format that will produce the list. The first line produced by the format will appear as the title of the list box during data entry. For example:

```
choice::'Languages'/'English'/'French'/'German'/'Spanish'
```

The values here are all unconditional literals. 'Languages' will be taken as the title and 'English', 'French', 'German', and 'Spanish' will be the choices displayed as shown below:



You can allow the user to select more than one item from the list by using the *multi* keyword and *repeat* puts each value into a separate occurrence of the field:

```
choice:multi:repeat::'Languages'/'English'/'French'/'German'/'Spanish'
```

You can specify that each value is enclosed in angle brackets:

```
choice:multi:<>::'Languages'/'English'/'French'/'German'/ 'Spanish'
```

You will then need to index the field with Indexing Technique 2.

If you follow a standard format for bibliographic records, you may wish to hold the language as a code but display the full name in the pick-list. This can be done using the *firstdescribe* keyword: each entry consists of its name followed by its code:

```
choice:firstdescribe::'Languages'/'French'/'fre'/'Portuguese'/'por'/
'Spanish'/'spa'
```

You can keep the list in a separate text file and give the file name here. Each line in the text file is treated as an entry in the list.

```
choice:files::'Languages'/'langs.txt'
```

The possibilities described here can be combined, i.e. you can use several keywords separated by colons and the order does not matter, but remember to put *two* colons before the list values or file name.

17.2 Pick Lists in J-ISIS

Pick lists can be defined in files called *worksheetName.val* as well as in a file called *databaseName*.val where *worksheetName* stands for the name of the particular data entry worksheet for which the pick list definitions and validation rules will be used. The file *databaseName.val* will be used (if any) in case a particular worksheet named *worksheetName* has no file called *worksheetName.val*. This means that in J-ISIS you can define a default pick list and validation rule file and specific pick list and validation rule files for data entry worksheets. These files must be stored in the **/iwks** folder of the database. Pick lists can be defined at the field and/or subfield level (WinISIS has only field level pick lists). In the default *databaseName*.val file, pick lists and validation rules can be defined for all fields that need it, and only the pick lists defined in the selected worksheet will be available.

As their suffix implies, VAL files are mostly used for establishing validation rules, at both record and field level. The validation rules defined in *databaseName*.val *or worksheetName.val* files are applied when you click on the VAL button and before saving the record.
J-ISIS Pick Lists and validation rules use extensively the potential offered by ISIS(J-ISIS) Formatting Language. Literals are essential in writing ISIS Formatting instructions, here is below a brief reminder about literals:

Literals

A *literal* is a string of characters, enclosed between appropriate delimiters, which will be inserted *as is* in the output. Literals may be used, for example, to label fields.
Three types of literals may be specified:
conditional literals: define text which will be output only if the associated field is present in the record. If the associated field selector is a subfield command (e.g. v24^a), the text will be included only if the requested subfield is present in the field. If the associated field selector specifies a repeatable field, the text will only be included once, regardless of the number of occurrences of the field. Conditional literals are enclosed in double quotation marks (“”), e.g. **`"Title: "`**
repeatable literals: like conditional Literals, they define text to be output only if the associated field or subfield is present in the record. If the field is repeatable, however, the literal will be repeated for *each* occurrence of the field. Repeatable Literals are enclosed in vertical bars (|) e.g. **`|Author: |`**
unconditional Literals: define text which is always output regardless of the presence of fields. Unconditional Literals are enclosed in single quotes (‘), e.g. **`'Summary'`**. As unconditional Literals are always output as a single block of text (i.e. they cannot be split between two lines), their length should not exceed the line width otherwise they will be truncated. To output text exceeding one line, you should break it down into two or more Literals.

Escape Sequences

However, it is often necessary to include literals meta characters inside a literal.

The following J-ISIS formatting instruction:

```
" She said \"Hello!\" to me"V24^a
```

will produce

```
She said "Hello!" to me
```

if subfield ^a is present in field 24.

You need to escape single quote that are inside unconditional Literals as in the following validation rule for field 215:

```
215:if v215 : ' cm.' then 'Pas de point final, J-ISIS s\'en charge !' fi
```

The vertical bar can also be escaped inside a repeatable literal

17.3 How to define Pick-Lists

The Pick-list definitions must be entered directly in the *databaseName.val* or *worksheetName.val* file (using Notepad.exe, for example). The Pick-Lists can also be defined in the worksheet using the J-ISIS internal Data Entry worksheet editor (Field Level only) or the Advanced Worksheet Editor (Field and Subfield Level).

It's important to be aware that:

If Pick Lists are defined in the worksheet as well as in a *worksheetName.val* or *databaseName.val* file, then the Pick List defined in the *worksheetName.val* file or *databaseName.val* will be used.

For example in the CDS database /iwks folder, we have CDS.val, and CDS1.val files that define pick lists. Pick lists defined by CDS1.val will be used with CDS1 worksheet while other worksheet will used the pick lists defined in CDS.val
If you want to use the example pick list defined into the CDS1 worksheet:
`choice::'z pick list'/'z1'/'z2'/'z3'` for subfield V12^z
you should rename the CDS.val and CDS1.val files to filenames that doesn't match the database name or a worksheet names such as CDS-test.val and CDS1-test.val (or delete them)

Using an external editor, a field pick-list definition looks like the following line:

```
44:choice:<>:notype:multi::'my list'/'first'/'second'/'third'
```

A subfield pick list definition would look like this:

```
44^z:choice:<>:notype:multi::'my list'/'first'/'second'/'third'
```

Where:

44:	is the field tag to which the pick-list will be applied
44^z	would be the field tag followed by the subfield code to which the pick-list will be applied
choice:	is the pick-list command
<>:	automatically enclose all selected items between <>
notype:	prevents the user from typing in this field
multi:	the users may pick more than one item from the pick-list.
:'my list'/	is the title of the pick-list
'first'/…	is a CDS/ISIS format producing a number of lines. Each of these lines will become an element of the list of choice from which the user will pick one or more items.

PICK-LIST CONFIGURATION

A number of options allow a variety of different behaviours and effects for implementing pick-lists. The general syntax for defining a pick-list is as follows:

tag[subfield]:choice:[multi:][files:][add:][repeat:]
[noansi:][firstdescribe:][notype:][<>:][//:]:*format*

Commands in between square parenthesis are optional. The options are:

multi: the users may pick more than one item from the list.

files: each line will be considered as a file name and each line of such a file will be considered as a valid item. List files are by default in the same directory as PFTs (not tested with the DBNPAR yet). For example:

```
69:CHOICE:files::'My title'/'mycds.txt'/'mylist.txt'
```

add: new selected items' text will be added to the text already in the field.

noansi: consider the text produced by the format as a DOS text and performs the conversion.

repeat: build a new field occurrence from each selected item.

notype: prevents the user from typing in this field. As soon as the user types a key, the Pick-list window pops-up. However basic keys such as Enter, Escape and Shift-Enter are allowed.

<>: and // automatically enclose all selected items between <> or slashes. For example:

```
69:CHOICE:<>::'My keywords'/'Coastal Zones'/'Flood
control'/
will automatically produce: <Coastal Zones><Flood control>
```

firstdescribe: each entry is composed by a couple of lines. The first is what the user sees on the list. The second is what it will be really inserted in the field. This is useful to mask codes with human-readable descriptions.

```
69:CHOICE:firstdescribe:multi::'My keywords'/'Coastal
Zones'/'CZ'/'Brazil'/'BR'
```

The above format produces 5 lines:

```
My keywords          ← the window's title
Coastal Zones
CZ
Brazil
BR
```

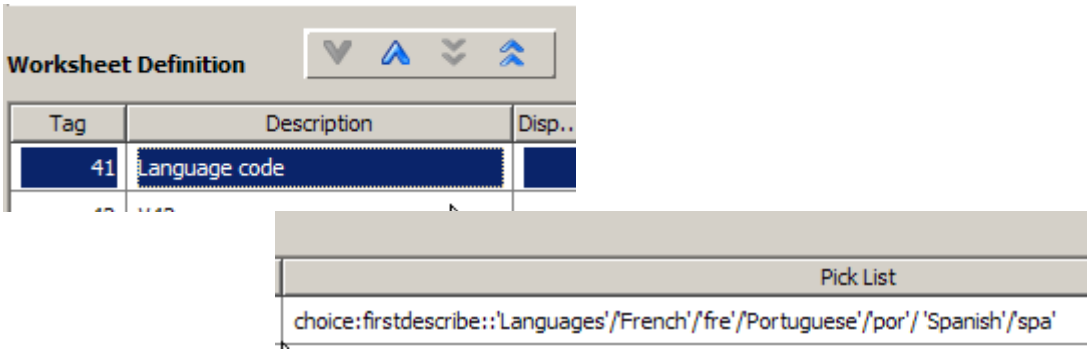
`format` The print format that defines the pick-list. The print format must produce a number of lines. Each of these lines will become an element of the list of choice from which the user will pick one or more items.

17.4 J-ISIS Worksheet Editors

J-ISIS **Edit** menu provides a ***Data Entry*** module and an ***Advanced Data Entry*** module. The Data Entry module allows entering data at the field level specifying explicitly the subfield delimiters, while the Advanced Data Entry module displays a hierarchical view of the worksheet fields and subfields that allows entering data at the subfield and field level.
Data is entered manually through a data entry interface specified by the user through a worksheet definition. **Data entry worksheet(s)** or **Advanced Data Entry worksheet(s)** are used to create and/or update the master records of the data base. J-ISIS **Edit** menu provides two specially designed editor to create these worksheets, Data ***Entry Worksheet*** module and ***Advanced Worksheet Editor***. The worksheet file xml format is compatible for both editor, and the ***Advanced Worksheet Editor*** allows entering further detailed information for data entry at the subfield level.

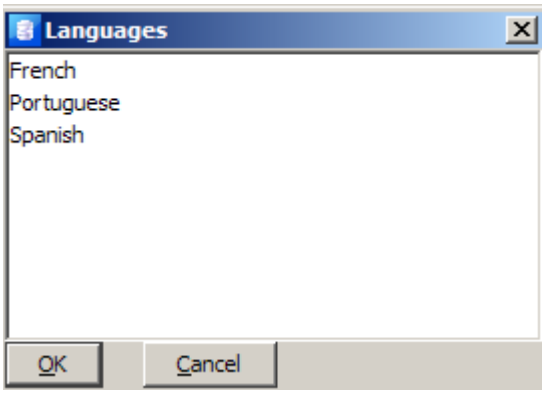
17.4.1 Field Pick List Definition in the Worksheet Editor

Selecting "Edit"->"Data Entry Worksheet" you can create or modify a data entry worksheet that can be further used in the "Data Entry" module. You just have access to fields but you can define a pick list as in the example below.



After saving the worksheet and opening the Data Entry module, you will see a Pick List button displayed in the field data entry panel.

Clicking on the Pick List button pops up a list selection dialog:



Double clicking on the “Spanish” item (or selecting it and clicking on “OK”) will fill the data box with the associated code:

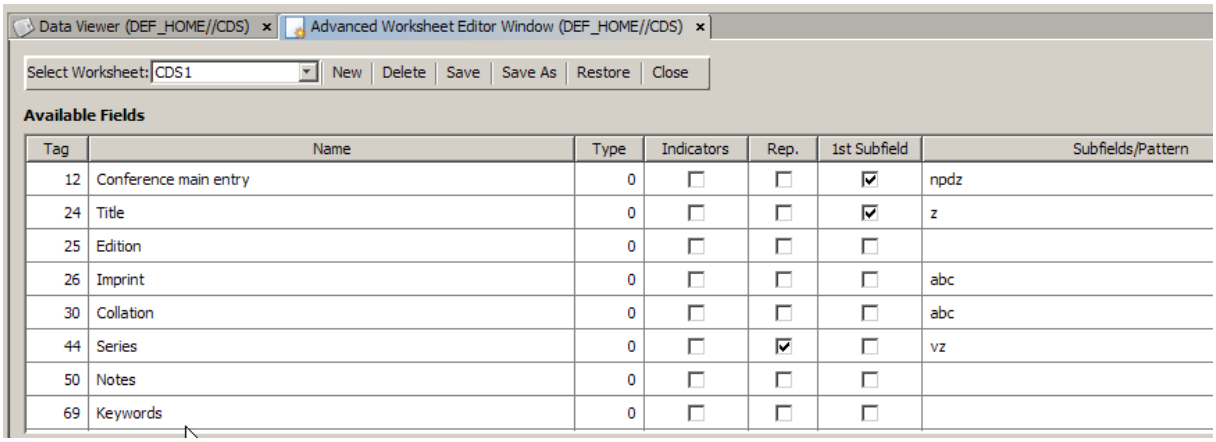
Field Pick List definition in a *worksheetName.val* or *databaseName.val* file

This Pick list could also be defined in a file called *worksheetName.val* or *databaseName.val*

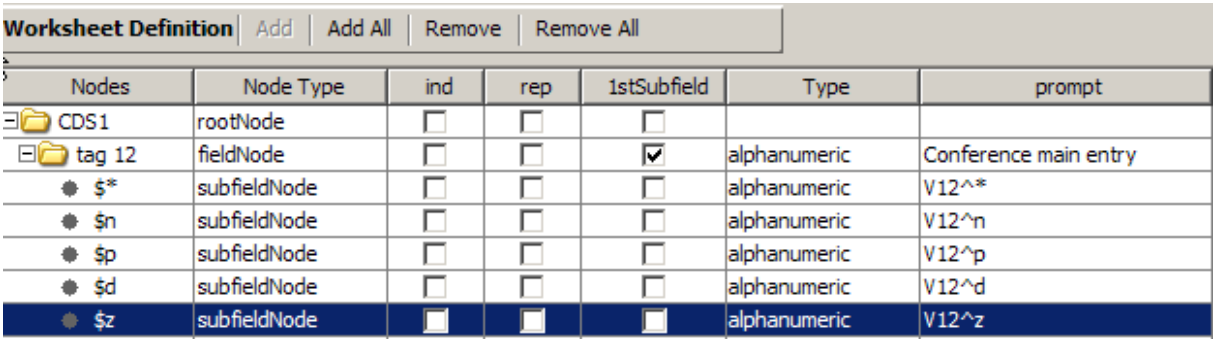
```
41:choice:firstdescribe::'Languages'/'French'/'fre'/'Portuguese'/'por'/
                  'Spanish'/'spa'
```

Subfield Pick Lists

You will need to enrich the worksheet with the advanced worksheet editor to be able to define subfield pick lists.
The advanced worksheet editor uses a Tree-Table layout and allows defining a worksheet that goes at the subfield level, define repetitive subfields and that may contain field indicators and implicit subfields.
A worksheet created with the *Data Entry Worksheet* Editor defines data entry fields at the field level. It doesn't include the subfields. The ***Advanced Worksheet Editor*** *allows* defining data entry fields at the subfield level. The worksheet xml file contains the worksheet field definitions as well as the subfield definitions if the worksheet was enriched using the ***Advanced Worksheet Editor.***



Tag	Name	Type	Indicators	Rep.	1st Subfield	Subfields/Pattern
12	Conference main entry	0	☐	☐	☑	npdz
24	Title	0	☐	☐	☑	z
25	Edition	0	☐	☐	☐	
26	Imprint	0	☐	☐	☐	abc
30	Collation	0	☐	☐	☐	abc
44	Series	0	☐	☑	☐	vz
50	Notes	0	☐	☐	☐	
69	Keywords	0	☐	☐	☐	



Nodes	Node Type	ind	rep	1stSubfield	Type	prompt
CDS1	rootNode	☐	☐	☐		
tag 12	fieldNode	☐	☐	☑	alphanumeric	Conference main entry
$*	subfieldNode	☐	☐	☐	alphanumeric	V12^*
$n	subfieldNode	☐	☐	☐	alphanumeric	V12^n
$p	subfieldNode	☐	☐	☐	alphanumeric	V12^p
$d	subfieldNode	☐	☐	☐	alphanumeric	V12^d
$z	subfieldNode	☐	☐	☐	alphanumeric	V12^z

prompt	defaultValue	displayControl	helpMsg	pickList
Conference main entry				
V12^*				
V12^n				
V12^p				
V12^d				
V12^z				choice::'z pick list'/'z1'/'z2'/'z3'

```
choice::'z pick list'/'z1'/'z2'/'z3'
```

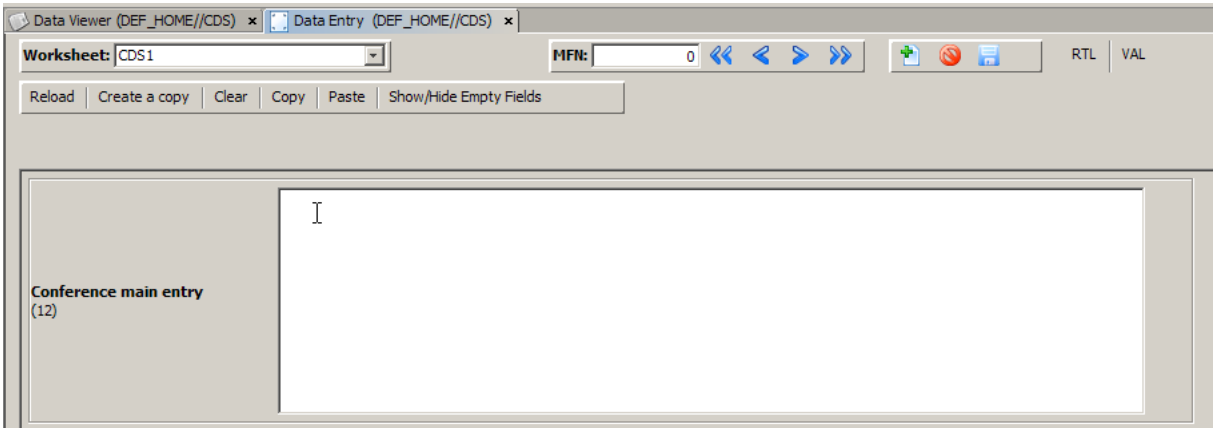
In the example above, a pick list is defined for subfield with code "z" using the Advanced Worksheet Editor.

Please note that the same worksheet named for example as “CDS1” can be edited either with the Worksheet Editor or the Advanced Worksheet Editor. The Advanced Worksheet Editor allows defining entry boxes for fields and subfields.

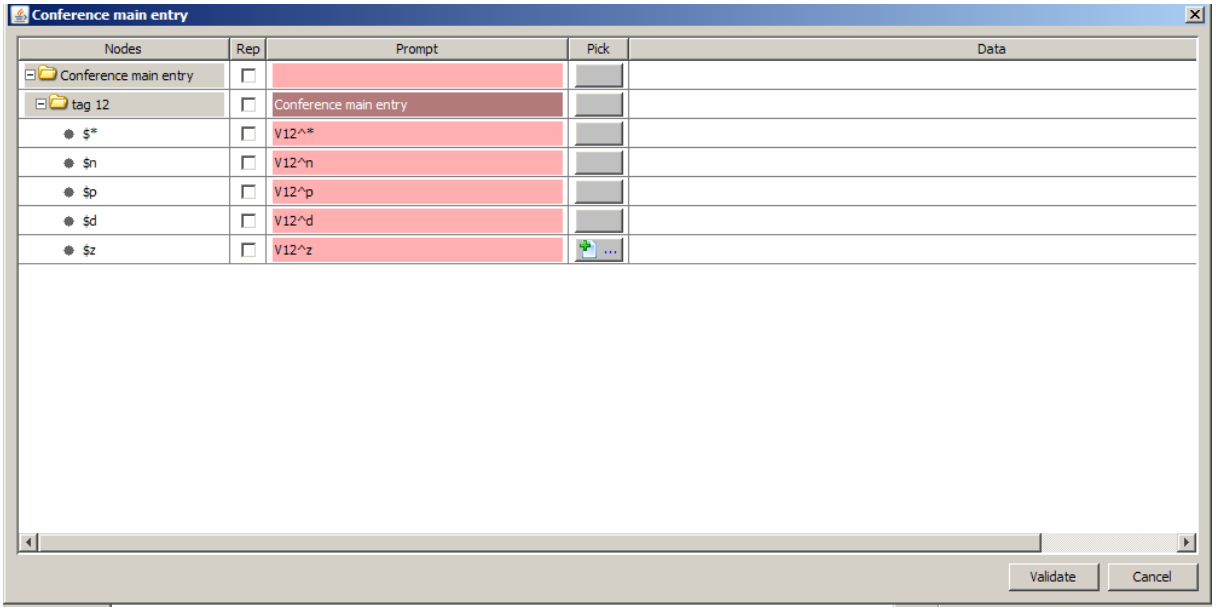
After saving the above worksheet with the subfield pick list, you will be able to access the subfield pick list in both data entry modules, “Data Entry” and “Advanced Data Entry”.

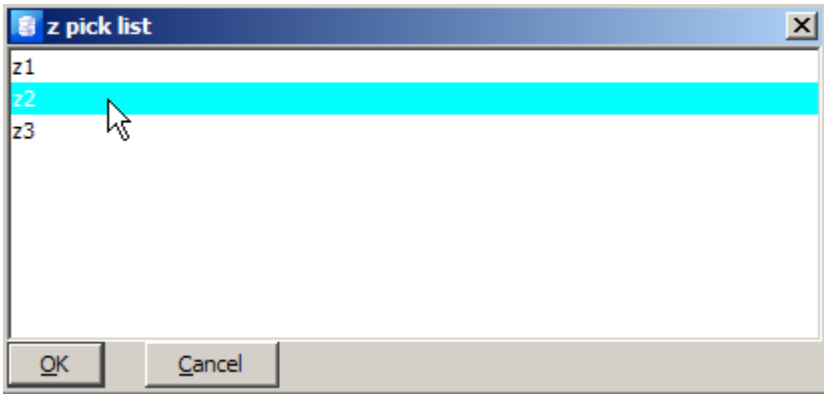
Accessing Subfields into the “Data Entry” module

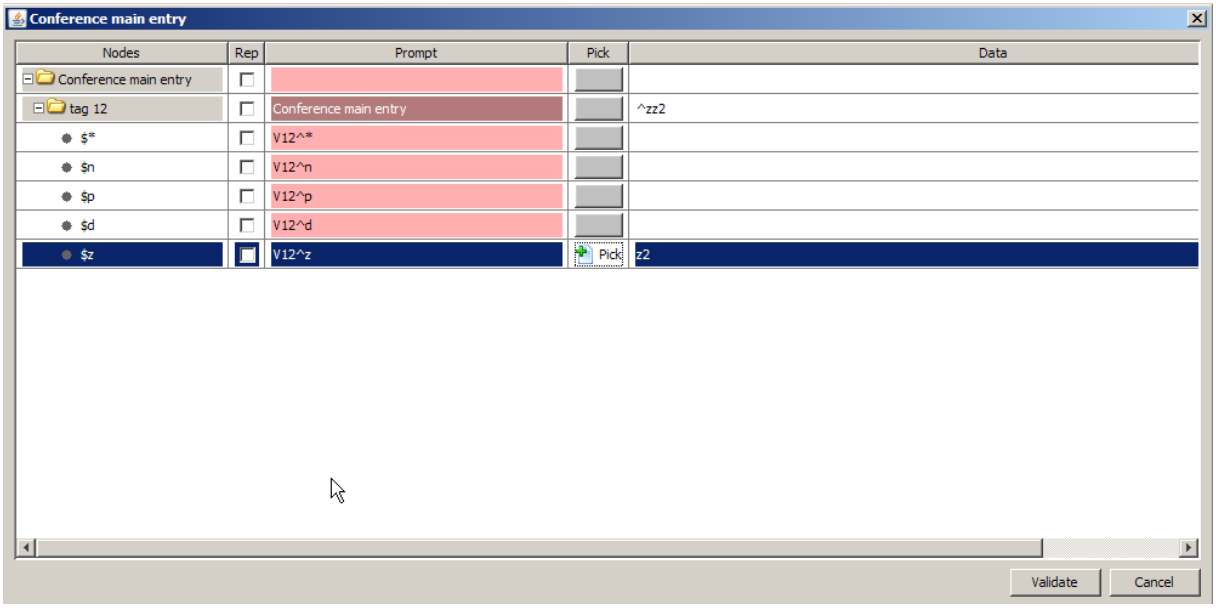
This is a new feature available since the 1st 2016 release. You can display a tree layout of the current selected field (where the cursor is positioned) by pressing “F10” key.



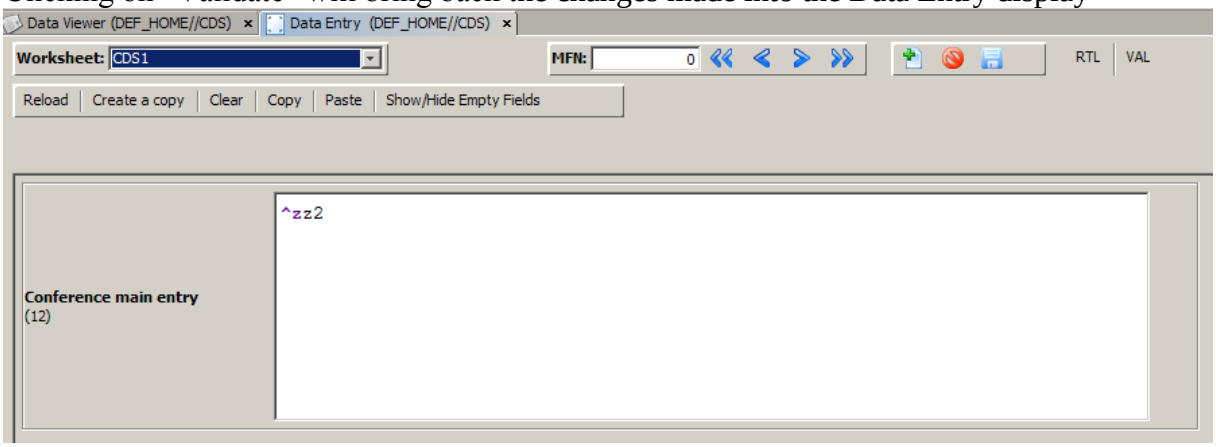
Pressing “F10” pops up a dialog with a hierarchical layout of the field as defined in the Advanced Worksheet Editor. A Pick List button is shown for subfield with code “z”



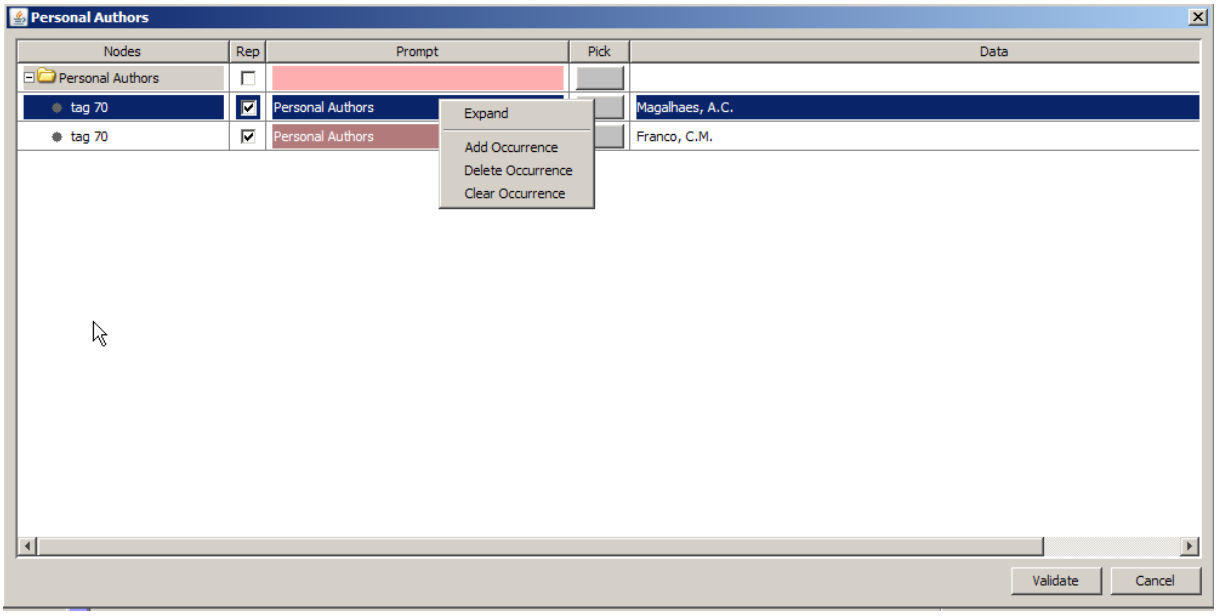


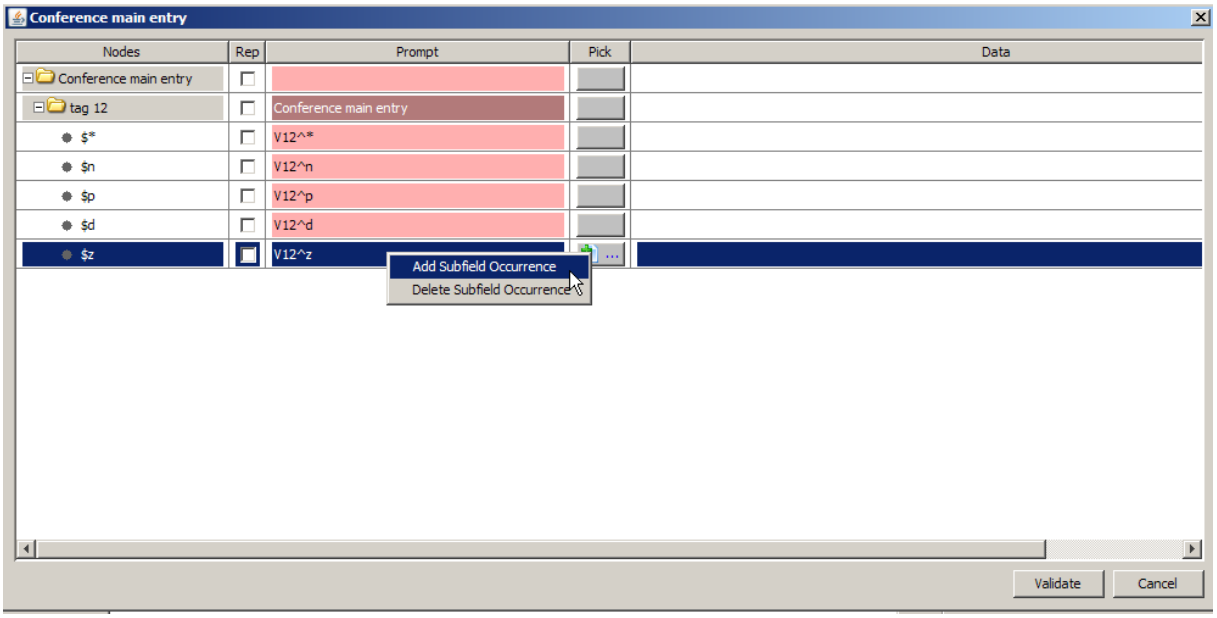


Clicking on “Validate” will bring back the changes made into the Data Entry display

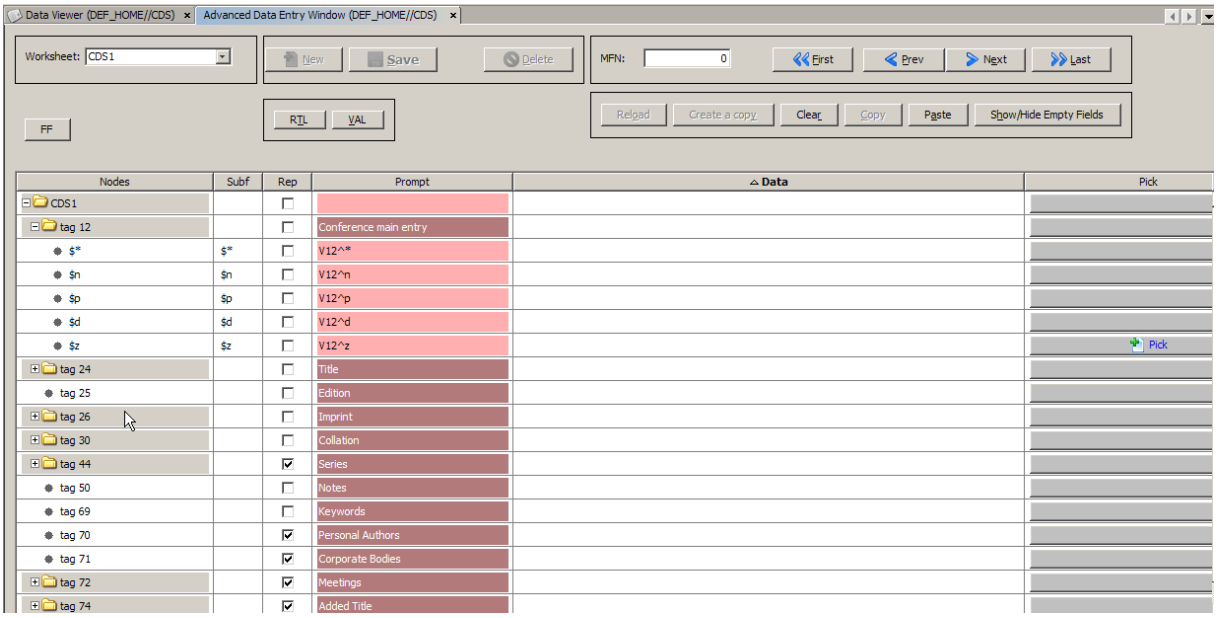


In case the field is repetitive, all occurrences will be displayed in the dialog tree layout. And clicking on the right mouse button when the cursor is on a field occurrence gives access to a context menu





Subfield Pick Lists into the “Advanced Data Entry” module

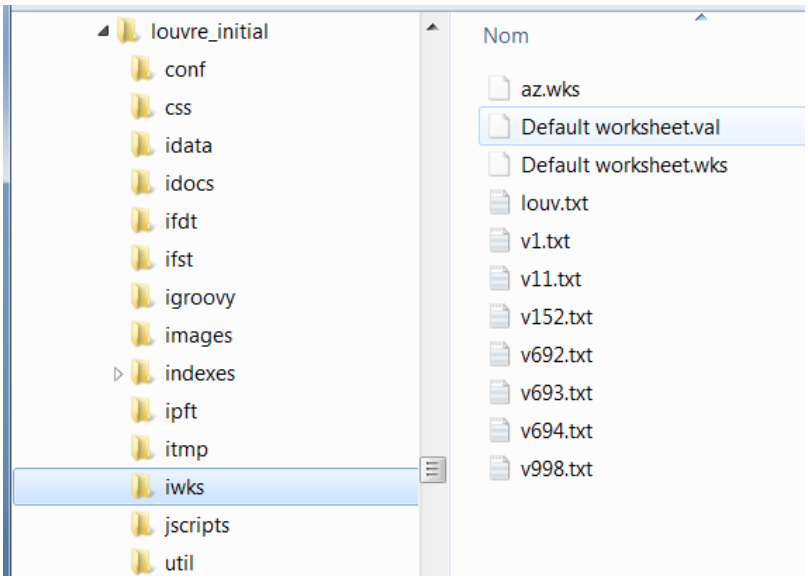


17.5 Examples

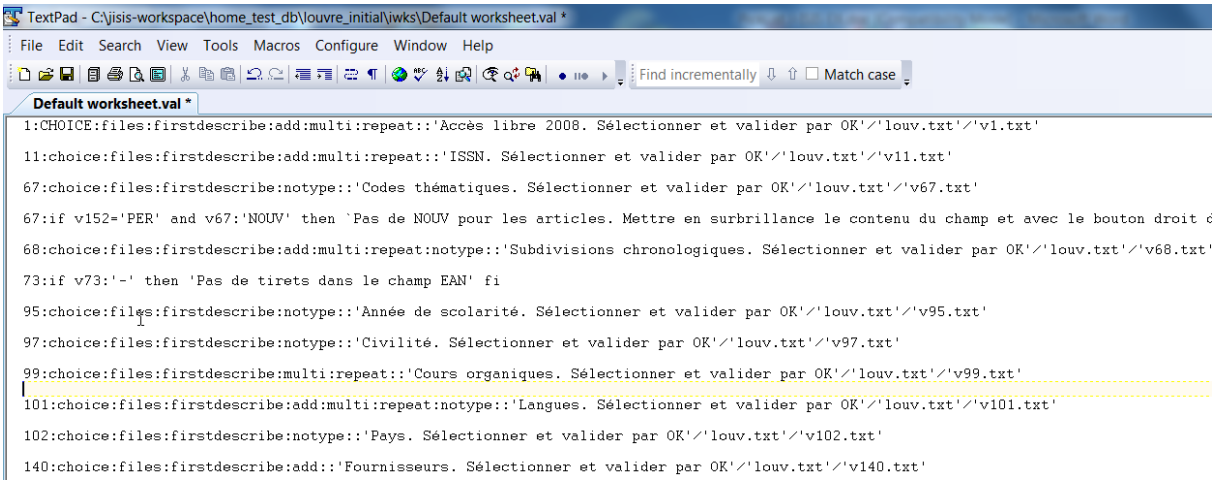
The database louvre_initial contains quite sophisticated pick list statements. The database folder /iwks is used to store worksheet definition files as well as pick list and validation files.

Important Notes:

- A pick list and validation file must have the same name than the worksheet file name that will used it.
- All files are UNICODE and should be saved using UTF-8 encoding



An extract of the file *louvre_initial.val* is displayed below:



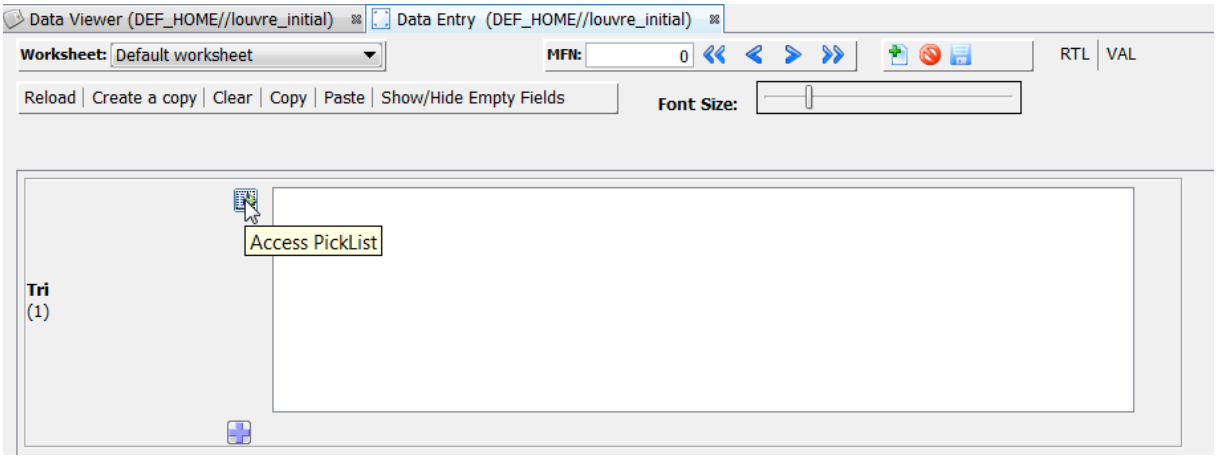
The file is edited with TextPad which is a good editor for building a *.val* file
If we analyse the 1st line:

1:CHOICE:files:firstdescribe:add:multi:repeat::'Accès libre 2008. Sélectionner et valider par OK'/'louv.txt'/'v1.txt'

We recognize the following options:

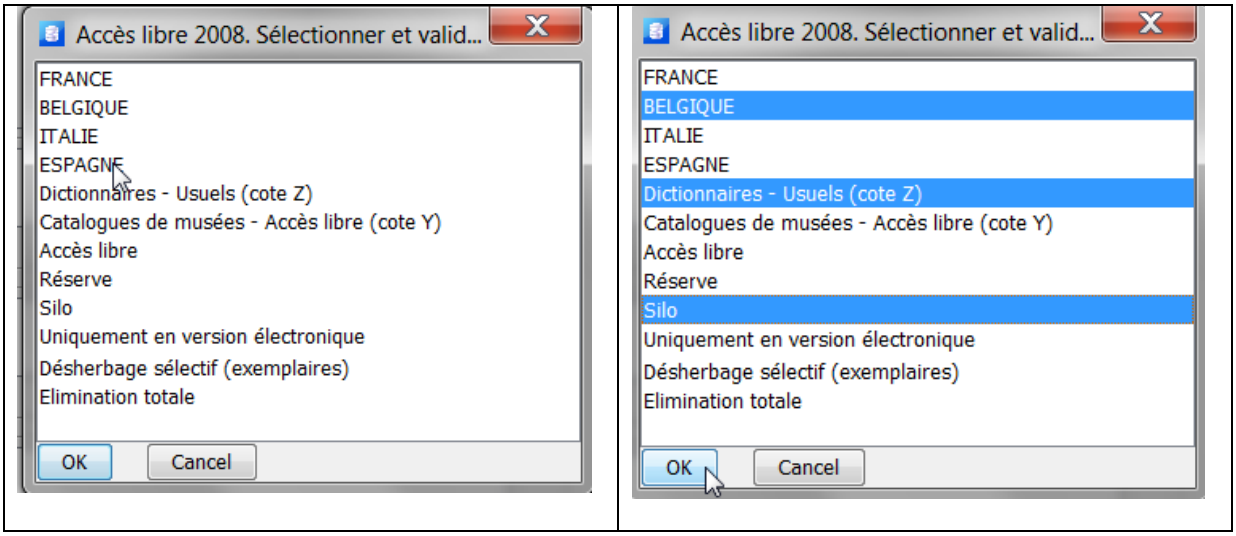
1:choice	This line defines a pick list for field associated with tag 1
:files	The pick list items are defined on files 'louv.txt' and 'v1.txt'
:firstdescribe	each entry is composed by a couple of lines. The first is what the user sees on the list. The second is what it will be really inserted in the field.
:add	new selected items' text will be added to the text already in the field
:multi	the users may pick more than one item from the list
:repeat	build a new field occurrence from each selected item
'Accès libre 2008. Sélectionner et valider par OK'	Title of the selection dialog
louv.txt	FRANCE FR BELGIQUE BE ITALIE IT ESPAGNE ES
V1.txt	Dictionnaires - Usuels (cote Z) USUEL Catalogues de musées - Accès libre (cote Y) ESSENTIEL Accès libre SUPER Réserve RESERVE Silo SILO Uniquement en version électronique EN LIGNE Désherbage sélectif (exemplaires) DESHERBAGE Elimination totale ELIMINATION

If now we look how it is interpreted at the Data Entry level, we can see that the field with tag 1and label 'Tri' has a Pick List button:

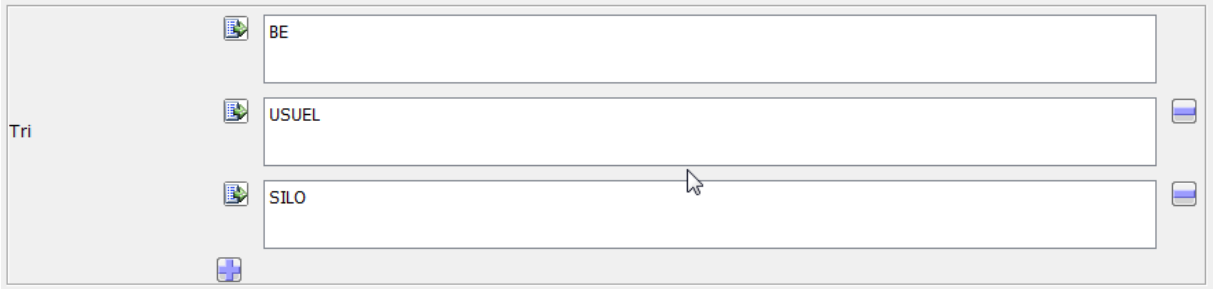


Clicking on the Pick List button will pop up a dialog with the elements defined through the *worksheetName.val* file for field with tag 1

As we have set the *multi* option, we can select several items using the standard Windows selection keys {mouse click} on first item to select and {ctrl mouse click} to select more than one item.



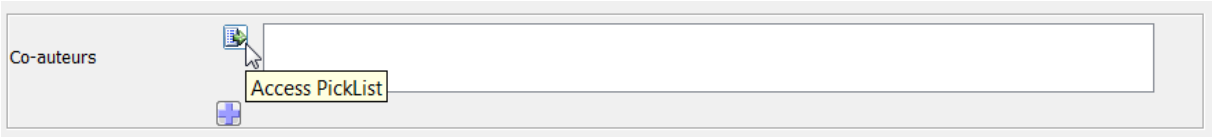
Clicking on OK will build a new field occurrence from each selected item , starting with the current occurrence if empty.

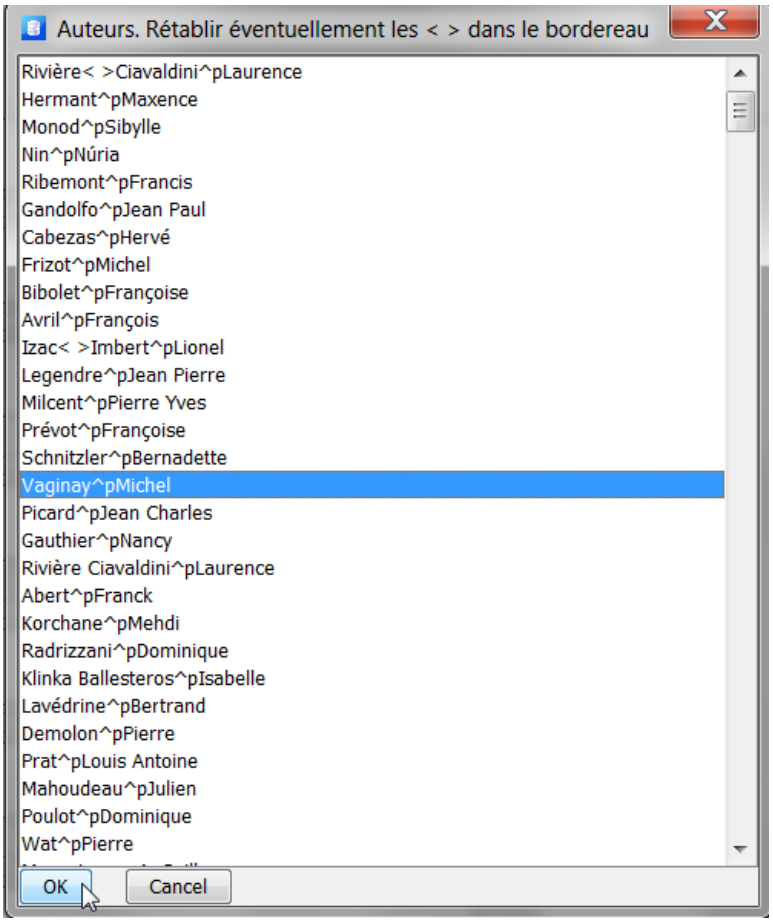


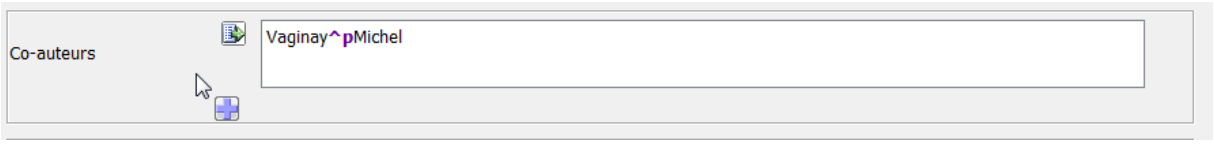
A more sophisticated example where we take the pick list items from another database:

```
701:choice:sort:add::'Auteurs. Rétablir éventuellement les < > dans le
bordereau'/ref->autor(lr->autor(('AUTEUR')),v2^a,,|^p|v2^b/)
```

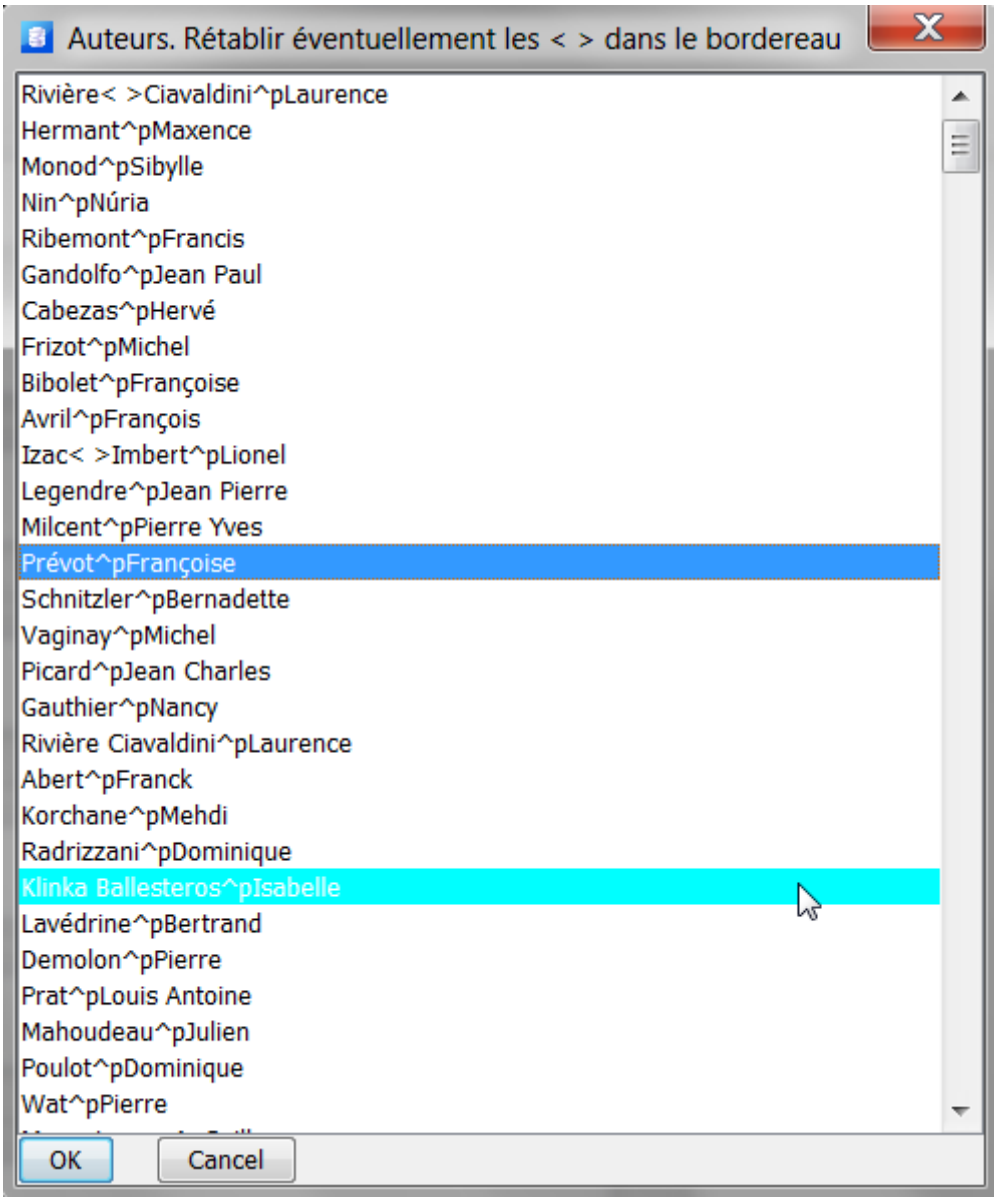
`701:choice`	This line defines a pick list for field with tag 701
`:sort`	ignored
`:add`	new selected items' text will be added to the text already in the field
`'Auteurs. Rétablir éventuellement les < > dans le bordereau'`	Title
`/ref->autor(lr->autor(('AUTEUR')),v2^a,,|^p|v2^b/)`	Format for building the pick list items







Finding an item in a long list is rather easy, jus begin to type characters and the selection will move to the first item that begins with these characters.
For example, typing pr will move to Prévot



17.3 How to define Validation Rules

The general syntax for defining field validation rules is as follows:

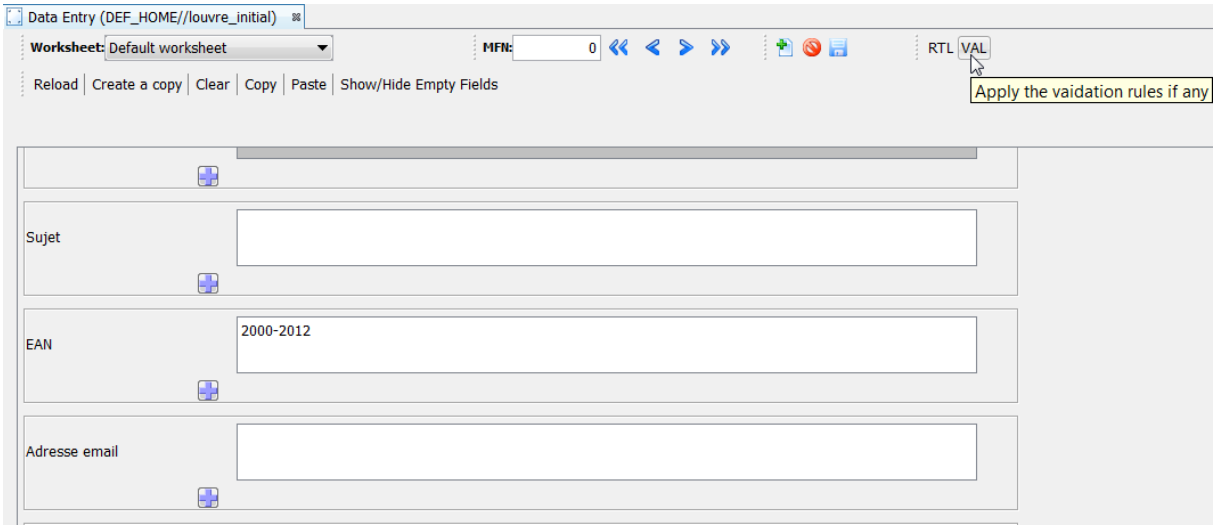
tag:format

Important Note: the 'begin:' and 'end:' record validation rules are not yet implemented

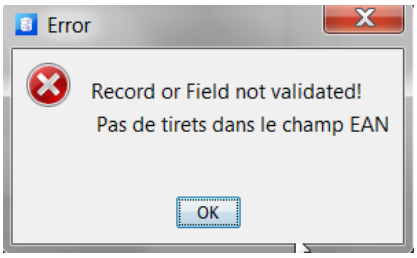
For example, the following rule test if field 73 contains a dash

```
73:if v73:'-' then 'Pas de tirets dans le champ EAN' fi
```

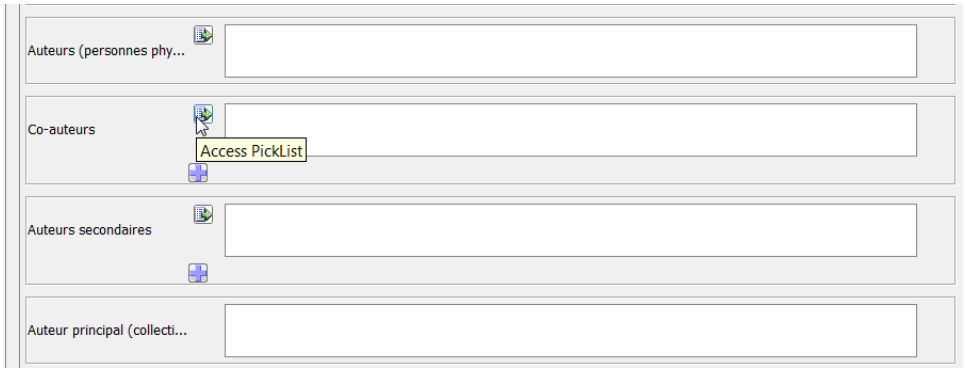
701:if p(V701) and a(V700) then 'Pas de co-auteur sans auteur principal' fi,
,if nocc(v701) > 2 and p(v700) then 'Plus de trois auteurs. Saisir les auteurs en v200^f' fi

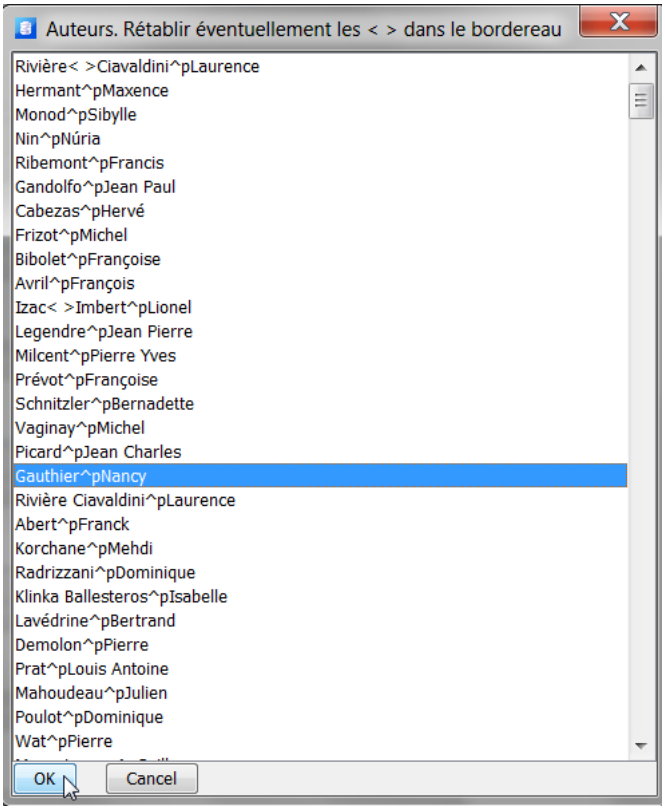


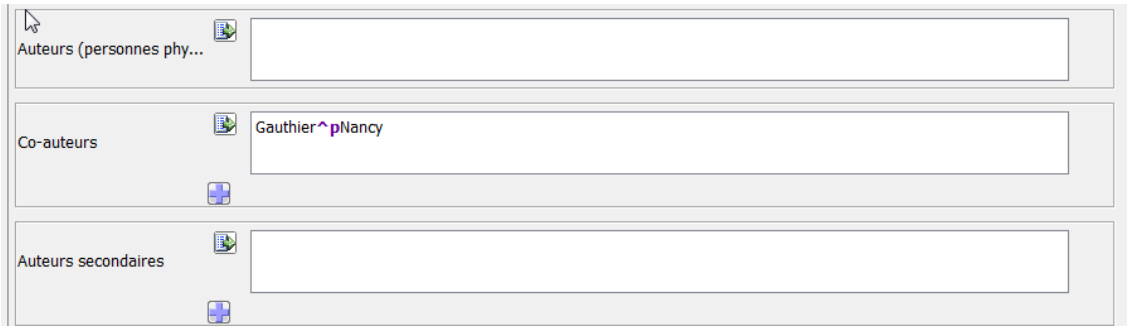
Tag 73 is associated to the EAN label, thus if we enter 2000-2012 in this field and click on the 'VAL' button, the following dialog will be displayed:

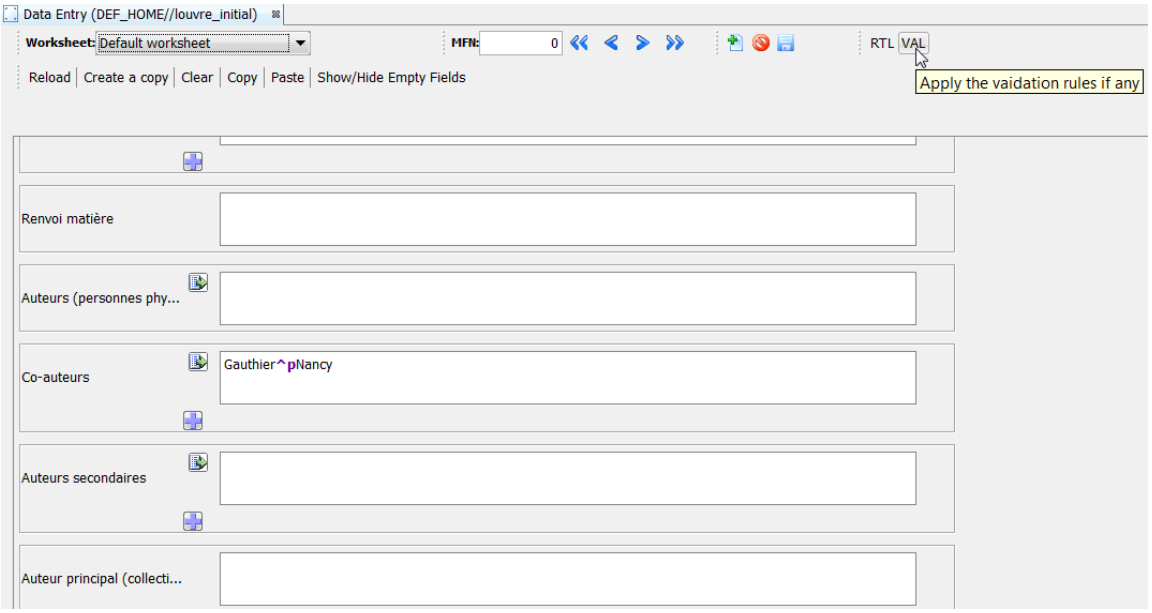


Selecting a 'co-auteur' through the pick list button as follow:




Auteurs. Rétablir éventuellement les < > dans le bordereau
Rivière< >Ciavaldini^pLaurence
Hermant^pMaxence
Monod^pSibylle
Nin^pNúria
Ribemont^pFrancis
Gandolfo^pJean Paul
Cabezas^pHervé
Frizot^pMichel
Bibolet^pFrançoise
Avril^pFrançois
Izac< >Imbert^pLionel
Legendre^pJean Pierre
Milcent^pPierre Yves
Prévot^pFrançoise
Schnitzler^pBernadette
Vaginay^pMichel
Picard^pJean Charles
Gauthier^pNancy
Rivière Ciavaldini^pLaurence
Abert^pFranck
Korchane^pMehdi
Radrizzani^pDominique
Klinka Ballesteros^pIsabelle
Lavédrine^pBertrand
Demolon^pPierre
Prat^pLouis Antoine
Mahoudeau^pJulien
Poulot^pDominique
Wat^pPierre
OK
Cancel


Auteurs (personnes phy...
Co-auteurs
Gauthier^pNancy
Auteurs secondaires


Data Entry (DEF_HOME//louvre_initial)
Worksheet: Default worksheet
MFN: 0
RTL VAL
Reload | Create a copy | Clear | Copy | Paste | Show/Hide Empty Fields
Apply the vaidation rules if any
Renvoi matière
Auteurs (personnes phy...
Co-auteurs
Gauthier^pNancy
Auteurs secondaires
Auteur principal (collecti...

And clicking on the 'VAL' button, we get the following dialogs: