Constants to XLIFF Files

To manage constants, you had to use 4DK# resources. 4D can now support constants in XLIFF files with rules describe below.
4D Pop Family should be updated shortly to provide tools for users constants migration and création of new ones in v13.

Description

- Constants loading mechanism

In previous version, 4D load the 4DK# resources from the 4D resources and then when a database is opened, the Plugins resources. This mechanism will continue to be supported in this version but is deprecated.

Now, 4D will load in addition the XLIFF constants definitions from the 4D folder 'Resources' and then when a database or a component is opened, the one available in the 'Resources' folder of the database and finally those found in the 'Resources' folder of the plugins package. Every time, search is performed at the first level of the 'Resources' folder and then in the current .lproj.

- XLIFF file architecture

Note : Constants are based on name. So the constant name can not be localize of course.

Here is an example of XLIFF transformation with explanation below.

I have a theme whose name is ‘my constants’ with 3 constants inside:

Previously I used a 4DK# with for example the ID 16001:
16001: ‘my constants’
Item 1: ‘first constant:-42:L’
Item 2: ‘second constant:Hello:S’
Item 3: ‘third constant:3.1415:R’

Now thrue a Xliff file, i twill appears like that :

```
<group restype="x-4DK#">
    <trans-unit id="1">
      <source>my constants</source>
    </trans-unit>
    <trans-unit id="2" d4:value="42">
      <source>first constant</source>
    </trans-unit>
    <trans-unit id="3" d4:value="Hello">
      <source>second constant</source>
    </trans-unit>
    <trans-unit id="4" d4:value="3.1415">
      <source>third constant</source>
    </trans-unit>
</group>
```

- restype="x-4DK#" indicates that the <group> element defines a constants theme.
- d4:value attribute specifies the value of the constant. "d4" is a namespace allowing to add attributes not provided by standard XLIFF. When the attribute d4:value is present inside a </trans-unit>, That's means this </trans-unit> defines a constant whose value is given by the attribute d4:value. The first </trans-unit> of the group is used for the theme name, so no d4:value attribute is defined.
The order of the </trans-unit> with d4:value attributes defines their index.
Note : the **ID attribute** (mandatory in XLIFF definition) is not used by 4D.

Be careful, constant type depends of the d4:value attribute.
If only numbers or minus signe are present, it will be evaluated as a longint.
If the value is bigger than a longint, or if it contains an exponent or a decimal, it will be evaluated as a real else it will be evaluated as a text.
You can set manually the type to affect by completing with following values: ":L", ":S" ou ":R" as previously.
Example:
d4:value="42:S" will provide a text constant whose value is "42"
If the value contains the ":" signe, you add to specify the type :
Example:

d4:value="mon:exemple:S"

- How to localize the theme name for a group of constants

Constants theme name can be localized. So you may want to define the name of the theme somewhere (in same or another XLIFF file).

Instead of including a <trans-unit> without an associated value in the group, a d4:groupName attribute has to be add in the <group> element to specify the resource XLIFF resname including the name to use.

Example:

```
<trans-unit id="1" resname="MyConstantsTitle">
<source>my constants</source>
<target>mes constantes</target>
</trans-unit>

<group restype="x-4DK#" d4:groupName="MyConstantsTitle">
    <trans-unit id="2" d4:value="42">
      <source>first constant</source>
    </trans-unit>
    <trans-unit id="3" d4:value="Hello">
      <source>second constant</source>
    </trans-unit>
    <trans-unit id="4" d4:value="3.1415">
     <source>third constant</source>
    </trans-unit>
</group>
```

- How to set the value of a constant of text type:

A constant can not be localize but can have different values following the language used. So you may want to define these different values somewhere (in same or another XLIFF file).

For example, the constants family to define the names of the months have different values following the language (e in French :"janvier, février,..." in English "january, february,...)

Instead of using the d4:value attribute, the d4:valueName attribute will be use to specify the resname of the XLIFF Resource including the value to use.

Example:

```
<trans-unit id="1" resname="MyConstantsTitle">
<source>my constants</source>
<target>mes constantes</target>
</trans-unit>
<trans-unit id="2" resname="MonthJanuary">
<source>january</source>
<target>janvier</target>
</trans-unit>
<trans-unit id="3" resname="MonthFebruary">
<source>february</source>
<target>février</target>
</trans-unit>


<group restype="x-4DK#" d4:groupName="MyConstantsTitle">
    <trans-unit id="4" d4:valueName="MonthJanuary">
      <source>kFirstMonth</source>
    </trans-unit>
    <trans-unit id="5" d4:valueName="MonthFebruary">
      <source>kSecondMonth</source>
    </trans-unit>
</group>
```

Particular constants

- Group with empty Constants
Target mut be empty :

```
<trans-unit id="1" d4:value="1">
    <source>my first constant</source>
    <target>my first constant</target>
</trans-unit
<trans-unit id="2" >
    <target/>
</trans-unit
<trans-unit id="3" d4:value="3">
```

```
    <source>my third constant</source>
    <target>my third constant</target>
</trans-unit
```

- Constants different between platforms
d4:includeIf attribute allows to define a trans-unit on a particular platform.
ex:
<trans-unit id ="1" d4:value="\" d4:includeIf="win">
<source>path delimiter</source>
<target>path delimiter</target>
</trans-unit>
<trans-unit id ="2" d4:value=":" d4:includeIf="mac">
<source>path delimiter</source>
<target>path delimiter</target>
</trans-unit>
On Windows, the first trans-unit will be take in account and on Mac-OS, it will be the second.
The same order number will be use during the constant traitment.

The d4:excludeIf attribute is the exact opposite. It allows not to define a trans-unit on a particular platform.

This mechanism (includeIf/excludeIf) will be common to all the xliff traitment and not only for constants.
This mechanism is also use for the xliff menu definition allowing to present Quit on Mac and Exit on Windows.