GUIDE TO THE XML REPORT FORMAT

This not a complete guide, just some notes on pitfalls and possibilities!

I have added some special features which are marked with an asterix (*) below!

Variables can be defined and used when constructing a report. Variable names follow common rules, shall consist of alfanumarical and underscore (_) characters. The value of a variable is retrieved with '$' preceding the variable name.

1 <setvar name="..." value="..." />

The value part can contain arithmetic expressions such as "2*($n-1)+1".
There are some special values that can be used (not in expressions!):
"@ID" will find the line "0 @xref@ xxx" of the current gedcom record and return "xref".
"@XXX" will find the first line containing the tag XXX (n XXX ...)
"@generation" will return the generation number, valid only when sortby="generation", see below.
”@format” will give the format of the report, ”PDF” or ”HTML” (or ”” if not given).
”@relation” will give the relation from the start person to the selected person.

The name part may also be a reference to a variable which then must contain a legitimate variable name, e.g. <setvar name="ddd" value="dval" /><setvar name="$ddd" value="25" /> will result in that variable dval will get the value 25. This is useless witout the addition below.

(*) The value can specify @$*nam* which returns the value of the variable the name of which is $*nam*. This addition makes it possible to retrieve the value assigned above through
<setvar name=”vvv” value=”@$ddd” /> vvv is set to 25.
Now pedigree charts can be generated in generalized way.

2 <if condition="..."> ... </if>

The condition can be specified almost as expected with the special trick that '>' must be written as 'GT', '<' as 'LT', '&&' as 'and', and '||' as 'or'. Spaces may be needed for separation. E.g. "$n GT 7" or "($n GT 0)and $x==0".

Repetions

A report will collect data from several individuals or families. I have found two useful constructs here, <Relatives ...> and <List ....>.

3 <Relatives *attr=value ...*> ... </Relatives>

This will create a loop to find relatives to an individual. In each iteration focus is set on one individual and his/her gedcom record is easily available. Example:

```
<SetVar name="sex" value="@SEX"/>
<SetVar name="mid" value="@ID"/>
<SetVar name="generation" value="@generation"/>
```

Attributes:

id	specifies the xref of the individual to start from	
group	can specify one of	
	child-family	Parents and siblings
	spouse-family	Spouses and children
	direct-ancestors	Direct line ancestors
	ancestors	Direct line ancestors and their families
	descendants	Descendants
	all	All
maxgen	will limit the number of generations to include, not valid for groups …-family	
sortby	will sort the result according to:	
	generation	by generation number, start person is generation 1, increased by 1 both up and down, not possible for ...-family or all
	NAME	by name
	BIRT:DATE	by birth date
	DEAT:DATE	by death date
	other	do not sort

4 <List *attr=value ...*> … </List>

This will create a loop with the individuals (or families) that pass the filters, possibly one to nine filters named filter1 to filter9.

Attributes:

list	individual or family
filter1	condition used to select individuals or families
…	
filter9	
sortby	will sort the result according to one of NAME, BIRT:DATE, DEAT:DATE, MARR:DATE or CHAN (change time).

Example of filters:

```
filter1="BIRT:PLAC CONTAINS $birthplace"
filter2="BIRT:DATE GTE $birthdate1"
filter3="BIRT:DATE LTE $birthdate2"
filter4="NAME CONTAINS $name"
```

5 <Facts ignore="$ignore" > ... </Facts>

For the current individual or family, loop over the tags not listed in $ignore. Focus is set on the found tag. Subrecords can be accessed directly, e.g."SOUR". Now variables @fact and @desc are available, @fact is the tag name and @desc is the rest of the line (n *TAG* ...). N.b. the same tag name can occur many times.

OUTPUT TO REPORT

When the focus is set on one individual, family or a fact it is time to create some output to the report. This should occur within a TextBox which can contain many Text areas. It seems that output can be generated without fulfilling this but don't rely on that!

In order to create text in the report the following xml elements are used.

6 <Style *attr*=*value* … />

Defines which font to use and its characteristics.

Attributes:

name	name of the style that is being defined
font	must be a recognized font name, 'dejavusans' is default
size	size to use, in points, e.g. "9"
style	"B" for bold, "I" for italics, "U" for underline, can be combined
color	not always recognized, most useful "#xxxxxx" to specify the color

7 <TextBox *attr*=*value* … > … </TextBox>

Attributes:

style	name of the style that is to be used
color	not always recognized, most useful "#xxxxxx" to specify the color

The contents in the text box can be any valid xml elements and output should be within text boxes, see below.

8 <Text *attr*=*value* … > ... </Text>

Attributes:

style	name of the style that is being defined
color	not always recognized, most useful "#xxxxxx" to specify the color

The contents can be pure text or <var ... > (see next paragraph) or other xml elements that print gedcom values, see below. N.B. space characters are significant and sometimes(??) new lines also. However tab characters in the beginning of a line are ignored together with the new line character!

9 <var var="*value*" />

This will output the text (or numerical value) of the variable with the name *value*. Here the name can be given in another variable, as <var var="$xyz" /> but note that expressions are **not** allowed.

Assuming that '<setvar name="xyz" value="dval" />' and '<setvar name="dval" value="25" />' are defined, then example will output the value of the variable dval which is 25. However, if a variable is not defined it will print the name of the variable. If dval had not been set the output would be dval!

value can also specify I18N functions:

- I18N::translate('*string*') which will find the translation of '*string*' to the language of choice.
- II18N::translate(*$variable*) which will find the translation of the value of the variable (*$variable*) to the language of choice.
- 18N::translateContext('*string*','*string*')
- I18N::number('*number*')

10 <GetPersonName id="..." select="*...*" fam_relation="n"/>

Attributes:

id	xref of the individual whose name to print
select (*)	Optional. Can be used to print the latest name ("latest") or a combination of latest name and the surname part of the birth (first) name: "Mary Johnsson b. Svensson" where 'b.' stands for 'born'. This is a common way to write names in Sweden!
fam_relation (*)	1 append relation of the current person to the main person to the name in the form (mfmf) where m=mother, f=father, s=son, d=daughter, x=sibling or partner 0 don't add anything, this is the default

Outputs the name of the current individual if id= is missing or is an empty string. It will show the first (birth) name of the individual whose xref is given as the id string unless another name part is specified with select=...

11 <GedcomValue tag="*value*" />

Will output the value of the subrecord of the record currently in focus. The tag may be a subrecord or a lower part if tag names are separated with ":", e.g. MARR:PLAC is valid if a family is selected to find the place where the marriage took place. Many other records can have tag="SOUR:DATA:TEXT" and so on.

12 <RepeatTag tag="XXX">

If the currently selected record has subrecords with tag XXX they are handled in a loop. Text can for example be produced with <Text><br />XXX=<GedcomValue tag="XXX" />. </Text>.

13 <Gedcom id="xid"> ... </Gedcom>

Can be used to switch focus to another gedcom record. Useful to be able to print parents of a selected induvidual, or children, or ... It can also select subrecords of the current record such as SOUR or NAME or …

14 <Image file="..." width="..." ln="..." top="…" left="…" />

If a gedcom record contains an image it can be included in the report using the directive 'file="@FILE"'. Other values of the file attribute exist but not specified here. If used inside a Textbox the image is placed at the position reached by previous text output, if any. The width attribute specifies the image width in points. The ln attribute specifies whether following text will come next to the image (ln="T") or on the next line (ln="N"). The position can be specified with top=… and left=…, not very useful inside text boxes.

15 <Line x1="" y1="" x2="" y2="" />

Will draw a line from point (x1,y1) to point (x2,y2). Lines can only be generated outside text boxes (but unsure of that)! Another restriction is that lines on html output must have x1>=x2!

Debugging help

A side-effect used for debugging is that when 'dump' is used in

```
<setvar name="dz"  value="@$dump" />
```

the variables defined so far are printed by ReportParserGenerate.php to a file named 'my-errors.log' at the webtrees top. The first $dval (if set, otherwise 0) entries are not printed. Example:

```
srcwidth='0'
srcleft='60'
sheight='15'
width='0'
width1='0'
width2='0'
explain='Förfäder i direkt linje samt deras familjer'
dump='dval'
dval='25'
```

A special attribute can be added to specify which variable value to print:

```
<setvar name="dz" value="@$dump"  dumpvar="xyz" />
```

will print

var: xyz = *value*

to the file 'my-errors.log'. With attribute dumpvar="gedcom" the current gedcom record will be printed.