

# LIST TEMPLATE FORMAT V1.01

## 09/02/98

**Bold\_string** Must be entered as shown  
UPPER\_CASE\_STRING Numerical value  
*italic\_string* Syntax descriptor

### Special syntax descriptors:

(( *item* )) Complex syntax item  
[[ *item* ]] Optional item  
*item1* || *item2* Select one of the items  
... Repeat the previous item any times  
*item1* ::= *item2* Define *item1* as *item2*

Keywords are case insensitive. Spaces, tabs, newlines can be put anywhere between the syntax items (comments and strings are special cases).

*char* ::= *any\_char\_except\_newlines*  
*comment* ::= (( // *char...* *newline* )) || (( /\* *any\_char...* \*/ ))

Comments can be put anywhere between the syntax items.

*string\_char* ::= *char\_except\_backslash* // \ \ || \n || \t || (( \ *char* ))

The \ \ means the backslash itself, \n means a newline, \t means a tab, otherwise the char after a backslash means the char itself.

*string* ::= (( " *string\_char...* " )) || *string1* *string2*

The second form is available, if a string is too long, because the string itself cannot contain a newline.

*template* ::= { *versionItem* [[ *debugItem* ]] [[ *listFormatItem* ]]  
[[ *listTypeItem* ]] [[ *unitItem* ]] [[ *pageItem* ]]  
[[ *lineGapItem* ]] [[ *alwaysItem* ]] [[ *userTextItem* ]]  
[[ *headerItem* ]]  
[[ *footerItem* ]]  
[[ *repetitiveItem* ]]  
[[ *repetitiveItem* ]]  
}

*versionItem* ::= **Version** NUMBER

This obligatory item is compared with the version number stored in the application.

*debugItem* ::= **Debug**

If this item appears, then the list engine sends out visible markers around the frames, and the format texts instead of the real content of the fields.

*listFormatItem* ::= **Listformat** (( **Book** // **Spreadsheet** ))

The default value is book.

*listTypeItem* ::= **Listtype** (( **Zone** // **Object** // **Component** ))

The default value is object.

*unitItem* ::= **Unit** ( ( **mm** // **cm** // **inch** ) )

The default value is mm. This is the paper unit. It sets the page size and the line gap. The default of the page size is 297 and 210 if the unit is mm, 29.7 and 21 if the unit is cm, 11 and 8 if the page size is inch. The default of line gap is 1 if the unit is mm, 0.1 if the line gap is cm, 0.03125 (1/32) if the unit is inch.

*pageItem* ::= **Page** ( ( **HEIGHT** , **WIDTH** ) ) // **Printer** // **Plotter**

0 < **HEIGHT**. 0 < **WIDTH**. The page size the given size, or the size of the usable area of the selected page in the Page Setup dialog or the Plot Setup dialog. This is the default value, but notice the unit can redefine this value.

*linegapItem* ::= **Linegap** **GAP**

The default value is 1, if the unit is mm, 0.1, if the unit is cm, 0.03125 (1/32), if the unit is inch.

*userTextItem* ::= **Usertext** [ **INDEX** ] *string* // **Name** *string*

1 <= **INDEX** <= 5. Define the value of a user text. The first version is direct, the second is the content of a text file. The default of string is " ".

*alwaysItem* ::= **Always**

If this keyword is present, then all the level records are always written out regardless of the order number of the parameter has changed. If this keyword is not present and the nth parameter has changed, then only the nth, (n + 1)th, ... are written out.

*headerItem* ::= **Header** // **Firstpageheader** { *levelTotalDesc* }

*footerItem* ::= **Footer** // **Firstpagefooter** { *levelTotalDesc* }

*repetitiveItem* ::= **Level** [ **INDEX1** [ [ , **INDEX2** ] ] ] // **Total** [ **INDEX1** [ [ , **INDEX2** ] ] ] // **Headline** [ **INDEX1** [ [ , **INDEX2** ] ] ] // **Based On** ( ( **Level** [ **INDEX3** ] // **Total** [ **INDEX3** ] // **Headline** [ **INDEX3** ] ) ) ] ] [ [ **For All** ] ] { *levelTotalDesc* }

1 <= **INDEX1** <= **INDEX2** <= 60. 1 <= **INDEX3** <= 60. Two items of the same type and index is not allowed in a template. **INDEX3** must be an already defined item. If for all, then use this item for all secondary index.

*levelTotalDesc* ::= [ [ *frame* ] ] [ [ *textStyle* ] ] [ [ *format* ] ] [ [ *options* ] ] [ [ *picture* ] ] [ [ **Columns** { *column...* } ] ] [ [ *graphicItem* ] ]...

Frame is in relative paper coordinates. If both left and right values are zero, level or total does not appear. If both top and bottom values are zero then the item appears at the top of the

first free space. If top and bottom values are equal, but positive, then the item appears with this offset from the first free space.

<i>column</i>	::=	{ [[ <i>frame</i> ]] [[ <i>textStyle</i> ]] [[ <i>format</i> ]] }	An optional field in a level/total.
<i>picture</i>	::=	<b>Picture</b> { [[ <i>frame</i> ]] [[ <i>origin</i> ]] [[ <i>name</i> ]] [[ <i>pictureScale</i> ]] [[ <i>picturePosition</i> ]] }	An optional field in a level/total.
<i>frame</i>	::=	<b>Frame</b> LEFT [[ , RIGHT [[ , TOP [[ , BOTTOM ]] ]] ]]	-Paperwidth <= LEFT <= Paperwidth, -Paperwidth <= RIGHT <= Paperwidth, -Paperheight <= TOP <= Paperheight, -Paperheight <= BOTTOM <= Paperheight. If the LEFT or RIGHT are negative then the real left or right will be the Paperwidth plus LEFT or RIGHT, otherwise LEFT or RIGHT. If the BOTTOM or TOP are negative then the real bottom or top will be the Paperheight plus BOTTOM or TOP, otherwise BOTTOM or TOP. The real left must not be less than the real right. The real top must not be greater than the real bottom. The default of the RIGHT is the value of the LEFT. The default of the BOTTOM is the value of the TOP. The default of the TOP is zero. If the entire frame is missing, then the default is 0,0,0,0.
<i>textStyle</i>	::=	<b>Text</b> PEN , [[ <i>string</i> , [[ SIZE , [[ <i>style</i> , [[ <i>justification</i> , [[ <i>truncating</i> ]] ]] ]] ]]	The default of PEN is 1 (1 <= PEN <= 99). The string is the font name, its default is " Times ". The default of SIZE is 9 (4 <= SIZE).
<i>style</i>	::=	<b>Plain</b> // (( <i>styleItem</i> [[ + <i>styleItem</i> ]])... )	
<i>styleItem</i>	::=	<b>Bold</b> // <b>Italic</b> // <b>Underline</b> // <b>Outline</b> // <b>Shadow</b> // <b>Condensed</b> // <b>Extended</b>	The default is plain.
<i>justification</i>	::=	<b>Left</b> // <b>Right</b> // <b>Centered</b>	The default is left.
<i>truncating</i>	::=	<i>truncateItem</i> [[ + <i>truncateItem</i> ]]	
<i>truncateItem</i>	::=	<b>Truncated</b> // <b>Multiline</b>	The default is truncated. If the output text is too long, then it is truncated to one line with three dots, break into multiple lines (if the frame is big enough), or truncate each long lines.
<i>format</i>	::=	<b>Format</b> <i>string</i>	The default is " ". The string may contain special format items see below

*options* ::= **Options** (( **None** // (( *optionItem* [[ , *optionItem* )) ))

*optionItem* ::= (( **Pagebreak** [[ *positionItem* [[ + *positionItem* ]] ]] // **Only** ))

*positionItem* ::= **Before** // **After**

The default is none. Pagebreak before (default) and/or after the level. Pagebreak is not allowed in a header or footer. Only one item of a group will be shown if a parameter changes. A group is a series of levels/headlines/totals, all of them has an only option, but the previous and following levels/headlines/totals are not existing or have no only options.

*origin* ::= **Origin** (( **None** // **File** // **Preview** // **Data** ))

The default is none. It draws an empty frame with an X, uses a PICT (GIFF,...) file, it uses the preview of a library part, it uses the picture coming from the processed data, it uses the INDEXth encapsulated picture of a library part. 1 <= INDEX.

*name* ::= **Name** *string*

The default is " ". The string is the name of the PICT (GIFF,...) file or the library part.

*pictureScale* ::= **Scale** *SCALE* // **Auto**

SCALE > 0. The default is auto. The scale factor of the pictue. Auto means the optimal scale for the given frame.

*picturePosition* ::= **Position** (( **Lefttop** // **Top** // **Righttop** // **Left** // **Centered** // **Right** // **Leftbottom** // **Bottom** // **Rightbottom** ))

The default is Lefttop. The position of the picture within the frame.

*graphicItem* ::= *rect* // *line* // *circle* // *arc*

*rect* ::= (( **Rect** *LEFT* , *RIGHT* , *TOP* , *BOTTOM* [[ , *PEN* ]] )) // (( **Roundrect** *LEFT* , *RIGHT* , *TOP* , *BOTTOM* , *RADIUS* [[ , *PEN* ]] ))

If LEFT, or RIGHT is positive, then it is an absolute coordinate on the paper. If it is negative, then the program uses the sum of the given value and the paper width. If TOP, or BOTTOM is positive, it is measured from the top of the level or total field. If it is neagtive, then the program uses the sum of the given value and the bottom of the level or total field. 1 <= PEN <= 99.

*line* ::= **Line** *LEFT* , *RIGHT* , *TOP* , *BOTTOM* [[ , *PEN* ]]

If LEFT, or RIGHT is positive, then it is an absolute coordinate on the paper. If it is negative, then the program

uses the sum of the given value and the paper width. If TOP, or BOTTOM is positive, it is measured from the top of the level or total field. If it is neagive, then the program uses the sum of the given value and the bottom of the level or total field.  $1 \leq \text{PEN} \leq 99$ .

*circle* ::= **Circle** CENTERX , CENTERY , RADIUS [[ , PEN ]]

$0 \leq \text{RADIUS}$ .  $1 \leq \text{PEN} \leq 99$ . If CENTERX is positive, then it is an absolute coordinate on the paper. If it is negative, then the program uses the sum of the given value and the paper width. If CENTERY is positive, it is measured from the top of the level or total field. If it is neagive, then the program uses the sum of the given value and the bottom of the level or total field.

*arc* ::= **Arc** CENTERX , CENTERY , RADIUS , ALPHA , BETA [[ , PEN ]]

$0 \leq \text{RADIUS}$ .  $0 \leq \text{ALPHA} < 360$ .  $\text{ALPHA} \leq \text{BETA} \leq \text{ALPHA} + 360$ .  $1 \leq \text{PEN} \leq 99$ . If CENTERX is positive, then it is an absolute coordinate on the paper. If it is negative, then the program uses the sum of the given value and the paper width. If CENTERY is positive, it is measured from the top of the level or total field. If it is neagive, then the program uses the sum of the given value and the bottom of the level or total field.

### Items of the format texts

*index* ::= [ (( **Current** // INDEX1 )) [[ , (( **Current** // INDEX2 )) ]]

$1 \leq \text{INDEX1} \leq 60$ ,  $1 \leq \text{INDEX2}$ . The default of INDEX1 is 1. The default of INDEX2 is current, if 'for all' condition is active for the level (see repetitiveItem above), or 1. Current for INDEX1 means the number of level or total, which contains the index, current for INDEX2 means the current secondary index or 1.

*index2* ::= [ INDEX ]

$1 \leq \text{INDEX} \leq 5$ .

*item* ::= (( # // ^ )) (( identifier // (( ( expression ) )) [[ : WIDTH [[ . PREC ]]

$0 < \text{WIDTH}$ .  $0 \leq \text{PREC} < \text{WIDTH}$ . The default value for the WIDTH is the width of the input data, for the PREC is 0. The item will be replaced in the output text with the current value, if the current value is too long, it will be truncated to WIDTH, if too short, spaces will be added. If an item has no meaning at the current environment, then it will be replaced with WIDTH pieces of spaces. If item begins with a ^ character, and it is numeric type, then the smallest greater integer will be used (the ceiling function).

*identifier*

```
::= Page // Date // Time // Project // Parameter index //
UserText index2 // Quantity index // Unit index //
Value index // Total index // Keycode index //
Keyname index // Code index // Number index //
Name index // Totalnumber // Current // Current2
```

Page is the current page number.

Date is the day, on which the listing has started. If PREC is 0 (the default), then date is short (like 1/23/98), if PREC is 1, then the date is abbreviated (like Fri, Jan 23, 1998), if PREC is 2, then the date is long (like Friday, January 23, 1998).

Time is the time, when the listing has started. If PREC is 0 (the default), then time is without seconds (like 14:30), if PREC is 1, then the time is with seconds (like 14:30:00).

Project is the name of the current ArchiCAD project.

Parameter is the numerical value of the indexth computed value of the model, or the textual value of the indexth descriptor.

UserText is the content of the userText string or text file (see userTextItem above).

Quantity is the numerical value of quantity part of the indexth component.

Unit is the textual value of unit part of the indexth component.

Value is the numerical value of the indexth computed value multiplied by the quantity part of the component, or the textual value of the indexth descriptor.

Total is the sum of values with constant parameter[1], ..., parameter[index - 1] values.

Keycode is the code of the associated key if the parameter is a component or a descriptor.

Keyname is the name of the associated key if the parameter is a component or a descriptor.

Code is the code of the component or the descriptor.

Number is the number of the items, which has the same parameter[1], ..., parameter[index - 1] values.

Name is the name of the indexth parameter (name of the component or descriptor, fix parameter or prefix) .

Total number is the number of processed items.

Current is the index of the current level or total (1 <= value <= 60).

Current2 is the index of the sublevel (1 <= value). If 'for all' condition is not active for the level (see repetitiveItem above), then current2 is always 1.

**Notice:** If the format text is in a *headeritem* or in a *footeritem*, then identifiers referring to a parameter (**Parameter**, **Quantity**, **Unit**, **Value**, **Total**, **Keycode**, **Keyname**, **Code**, **Name**) can cause unexpected results.

*expression*

```
::= [[ + // - ]] [[ identifier // NUMBER ]] (( [[ + // - // * //
/ ]] [[ [ [ ^ ] identifier // NUMBER ]] ))...
```

If the identifier is textual, then zero value will be used. If there is a ^ character in front of the identifier, and it is numeric type, then the smallest greater integer will be used (the ceiling function). For example: If the expression is 5.5\*^parameter[1] + 1.7\* parameter [2], and parameter [1] is 3.3, parameter [2] is 1.5,

then the evaluated value is  $5.5*4+1.7*1.5=24.55$ .

### Example

```
{
// This is a sample list template text
// Created: 01/21/98

VERSION      1.00
LISTFORMAT   BOOK
LISTTYPE     OBJECT
UNIT         mm
PAGE         PRINTER
LINEGAP      1

HEADER {
  Frame      0, -6, 13, 18
  Text       1, "Times", 9, bold+italic, centered
  Format      "#project"

  Picture {
    Frame    -5.5, -1, 10, 23
    Origin   file
    Name     "Logo"
  }
  Columns {
    {
      Frame    0, -6, 18, 23
      Text     1, "Times", 9, italic
      Format    "Sample Listing #date:8 #time:5"
    }
  }
}

FOOTER {
  Frame      0, -1, -11, -1
  Text       1, "Times", 9, italic, centered
  Format      "- #page:3 -"
}

LEVEL [1] {
  Frame      0, -1
  Text       1, "Times", 12, bold
  Format      "Place: #parameter[1]"
  Options    pagebreak
}

TOTAL [4] {
  Frame      0, 110
  Text       1, "Times", 9
  Format      "#parameter[2] "
             "#value[5]x#value[6]"
  Columns {
    {
      Frame    -71, -41
      Text     1, "Times", 9, plain, right
      Format    "#value[3] #unit[3]"
    }
    {
      Frame    -41, -11
      Text     1, "Times", 9, bold, right
      Format    "#value[4] #unit[4]"
    }
    {
      Frame    45, -11, 6
      Text     1, "Times", 9, italic
    }
  }
}
```

```

    }
  }
  Picture {
    Frame      0, 40, 6, 56
    Origin     file
    Name       "Sample"
  }
}

TOTAL [2] {
  Frame      0, -71
  Text       1, "Times", 10, bold
  Format     "Total:"
  Columns {
    {
      Frame      -71, -41
      Text       1, "Times", 10, plain, right
      Format     "#total[3] #unit[3]"
    }
    {
      Frame      -41, -11
      Text       1, "Times", 10, bold, right
      Format     "#total[4] #unit[4]"
    }
  }
}

TOTAL [1] {
  Frame      0, -71
  Text       1, "Times", 12, bold
  Format     "Grand Total:"
  Columns {
    {
      Frame      -71, -41
      Text       1, "Times", 12, plain, right
      Format     "#total[3] #unit[3]"
    }
    {
      Frame      -41, -11
      Text       1, "Times", 12, bold, right
      Format     "#total[4] #unit[4]"
    }
  }
}
}

```

If the records coming from the model are:

Par[1]	Par[2]	Par[3]	Par[4]	Par[5]	Par[6]
"1st Floor"	"Double"	30	140000	100	100
"1st Floor"	"Double"	30	140000	100	100
"1st Floor"	"Single"	12	60000	60	80
"1st Floor"	"Single"	12	60000	60	80
"1st Floor"	"Single"	16	80000	80	80
"2nd Floor"	"Double"	30	140000	100	100
"2nd Floor"	"Single"	12	60000	60	80
"2nd Floor"	"Single"	12	60000	60	80
"2nd Floor"	"Single"	16	80000	80	80
"2nd Floor"	"Single"	16	80000	80	80

The list engine sends the following:

```

Level[1]
  Level[2]
    ...

```



Level[4]  
...  
Level[60]

for the 1st record.  
Nothing for the 2nd record (all parameters are equal).

Total[60]  
...  
Total[4]  
Total[3]  
(Level[1], if Always keyword would be present)  
Level[2]  
Level[3]  
Level[4]  
...  
Level[60]

for the 3rd record, because the parameter[2] has been changed.  
Nothing for the 4th record (all parameters are equal).

Total[60]  
...  
Total[4]  
(Level[1]  
Level[2], if Always keyword would be present)  
Level[3]  
Level[4]  
...  
Level[60]

for the 5th record, because the parameter[3] has been changed.

Total[60]  
...  
Total[4]  
Total[3]  
Total[2]  
Level[1]  
Level[2]  
Level[3]  
Level[4]  
...  
Level[60]

for the 6th record, because the parameter[1] has been changed.

Total[60]  
...  
Total[4]  
Total[3]  
(Level[1], if Always keyword would be present)  
Level[2]  
Level[3]  
Level[4]  
...  
Level[60]

for the 7th record, because the parameter[2] has been changed.  
Nothing for the 8th record (all parameters are equal).

Total[60]

...

Total[4]

(Level[1])

Level[2], if Always keyword would be present)

Level[3]

Level[4]

...

Level[60]

for the 9th record, because the parameter[3] has been changed.  
 Nothing for the 10th record (all parameters are equal).

Total[60]

...

Total[4]

Total[3]

Total[2]

Total[1]

for closing the list. Only Level[1], Total[1], Total[2], and Total[4] are defined, so the entire list contains

Level[1] (1st)  
 Total[4] (3rd, with 2nd record)  
 Total[4] (5th, with 4th record)  
 Total[4] (6th, with 5th record)  
 Total[2] (6th, with 5th record)  
 Level[1] (6th)  
 Total[4] (7th, with 6th record)  
 Total[4] (9th, with 8th record)  
 Total[4] (close, with 10th record)  
 Total[2] (close, with 10th record)  
 Total[1] (close, with 10th record)

Place: 1st Floor			
Double	100x10030 kg	140000 Ft	2 pieces
Single	60x80	12 kg	60000 Ft2 pieces
Single	80x80	16 kg	80000 Ft1 pieces
Total:	100 kg	480000 Ft	
Place: 2nd Floor			
Double	100x10030 kg	140000 Ft	1 pieces
Single	60x80	12 kg	60000 Ft2 pieces
Single	80x80	16 kg	80000 Ft2 pieces
Total:	86 kg	420000 Ft	
Grand Total:	186 kg	900000 Ft	