LIST TEMPLATE FORMAT

Graphisoft

Version 1.10 Copyright © Graphisoft 07/26/1999

Black_bold_string	Must be entered as shown.
GREEN_UPPER_CASE_STRING	Numerical value.
blue_italic_string	Syntax descriptor.
red_underlined_italic_string	Syntax descriptor with reference.

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 character after a backslash means the character itself.
string	*::=*	*((" string_char... ")) || ((string string))* The string defines a text, usually a name. The length of the string is not restricted, but in some usage there is a separate restriction. The second form is available, if a string is too long, because the string itself cannot contain a newline.
template	*::=*	{ *versionItem [[titleItem]] [[commentItem]] [[debugItem]] [[layoutScaleItem]] [[listFormatItem]] [[listTypeItem]] [[unitItem]] [[pageItem]] [[lineGapItem]] [[tileItem]] [[sequenceItem]] [[*

		cellGapItem]] [[*alwaysItem*]] [[*userTextItem*]]... [[*headerItem*]]... [[*footerItem*]]... [[*backgroundItem*]]... *repetitiveItem* [[*repetitiveItem*]]... } This is the entire template, which defines the format of the list.
versionItem	::=	**Version** NUMBER This obligatory item is compared with the version number stored in the application. If the defined version number is older or newer the number ArchiCAD recognizes, then the list template will not be interpreted.In other cases the method of interpreting can be slightly modified (for example see *pictureScale* statement). The version number will appear in Setup List Scheme dialog, when you select this template.
titleItem	::=	**Title** *string* The title string is the name of the layout. If the template is used in a layout, the layout editor will show this name. Otherwise it has no special meanings.
commentItem	::=	**Comment** *string* The comment string will appear in Setup List Scheme dialogs, when you select this template. If it is missing, there will be no comment in the dialogs.
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.
layoutScaleItem	::=	**Layoutscale** NUMBER This item defines the scale of the entire layout. This value will be used as the scale of the list window, and will be passed to the graphic macros, as global value (see also the picture with origin **Data**). The NUMBER is the dividend, so 100 means 1:100, 0.5 means 2:1 (1:0.5). The default value (in the case of missing **Layoutscale** command) is the scale of floor plan window of the ArchiCAD.
listFormatItem	::=	**Listformat** *((* **Book** || **Spreadsheet** *))* This item describes the form of the list. It is not used in the current version. The default value is **Book**.
listTypeItem	::=	**Listtype** *((* **Zone** || **Object** || **Component** *))* This item describes the source of the list. If the template used in a layout, the layout editor will use this value to select the

		layout type. Otherwise it is not used in the current version (the same template can be used for different list types). The default value is **Object**.
unitItem	::=	**Unit** *((* **mm** || **cm** || **inch** *))* This item describes the paper units used in the template. The unitItem must precede the pageItem and the linegapItem, if they are existing. The default value is **mm**. This item also sets the page size and the line gap. The default of the page size is 297 by 210 if the unit is **mm**, 29.7 by 21 if the unit is **cm**, 11 by 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** *))* This item defines the sizes of the used paper. The sizes are either the given values (HEIGHT > 0 and WIDTH > 0), or the sizes of the usable area of the selected page in the Page Setup dialog (**Printer**) or the Plot Setup dialog (**Plotter**). This item should follow the unitItem, if it is existing. About the default paper size see unitItem.
lineGapItem	::=	**Linegap** GAP This item defines the gap between the different parts of the list. This item should follow the unitItem, if it is existing. About the default gap see unitItem.
tileItem	::=	**Tile** NROWS *[[* **,** NCOLUMNS *]]* This item defines the number of the rows and the columns in a page (NROWS >= 1 and NCOLUMNS >= 1). The original pagesize (see pageItem) will be divided, and some gaps (see cellGapItem) will be used between the cells. The default value for both is 1. The record descriptions (see headerItem, footerItem, backgroundItem, and repetitiveItem) will refer only for one cell.
sequenceItem	::=	**Sequence** *((* **Byrow** || **Bycolumn** *))* This item defines the mode to browse among the cells. **Byrow** means to browse cells row&endash;by&endash;row, **Bycolumn** means to browse cells column&endash;by&endash;column. The default value is **Byrow**. If the number of columns (see tileItem) is 1, then only **Byrow** is allowed, if the number of the rows is 1, then only **Bycolumn**.
cellGapItem	::=	**Cellgap** HGAP *[[* **,** VGAP *]]* This item defines the gap between cells, if a page is divided (see tileItem). HGAP is the horizontal gap, VGAP is the

		vertical gap (HGAP >= 0 and VGAP >= 0). The default value for both is 0. If the number of columns (see tileItem) is 1, then HGAP must be 0, if the number of the rows is 1, then only VGAP must be 0.
userTextItem	*::=*	**Usertext [** INDEX **]** *((* *string* *||* *((* **Name** *string* *))))* These items define the values of textual variables, you can refer them later in any format statement. 1 <= INDEX <= 15. You can use only one userTextItem for each index. You can define it either directly, or as the content of a text file. In the latter case the string contains the file name and/or path with the restriction of the current OS. If you do not use path (this is the preferred), then ArchiCAD will look for the text file in any of the loaded library and near the application itself. The default values of the textual variables are "" (empty string).
alwaysItem	*::=*	**Always** If this keyword is present, then all the level/total/headline 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, ... records are written out. See Example.
headerItem	*::=*	*((* **Header** *||* **Firstpageheader** *)) ((* *string* *|| {* *levelTotalDesc* *}))* The **Firstpageheader** defines the format of the header on the first page of the list, the **Header** defines the format of the header of the list on all other pages (if **Firstpageheader** is missing, then on all pages). Only one definition is allowed of both types. The default header is the empty header. The definition contains the name of the pre&endash;defined record or the entire record description.
footerItem	*::=*	*((* **Footer** *||* **Firstpagefooter** *)) ((* *string* *|| {* *levelTotalDesc* *}))* The **Firstpagefooter** defines the format of the footer on the first page of the list, the **Footer** defines the format of the footer of the list on all other pages (if **Firstpagefooter** is missing, then on all pages). Only one definition is allowed of both types. The default footer is the empty footer. The definition contains the name of the pre&endash;defined record or the entire record description.
backgroundItem	*::=*	**Background** *((* *string* *|| {* *levelTotalDesc* *}))* The **Background** defines the format of the page background of all pages. Items in the background need no space on the page, the other items will be placed on the top of them. Only one definition of it is allowed. The default background is the

		empty background. The definition contains the name of the pre&endash;defined record or the entire record description.
repetitiveItem	*::=*	*((((* **Level** *||* **Total** *||* **Headline** *))* INDEX1 *[[* **,** INDEX2 *]]* **]** *[[* **Based On** *((* **Level** *||* **Total** *||* **Headline** *))* **[** INDEX3 **]** *]]* *[[* **For All** *]]* *((* *string* *|| {* *levelTotalDesc* *})))) ||* *((* **Sum [** INDEX1 **]** *string* *))))* 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. **Level** is written out for each instance. **Total** is written out after a parameter with smaller index has been changed (Total [1] is the grand total, Total [2] is written out, if the parameter 1 has been changed, etc.). **Headline** is written out before a parameter with smaller index has been changed (see **Total** for the indices). The definition contains the name of the pre&endash;defined record or the entire record description. **Sum** defines an expression, which behaves like a total value (it has different values on the different levels). The definition string contains an *expression*.
levelTotalDesc	*::=*	*[[* *frame* *]] [[* *textStyle* *]] [[* *format* *]] [[* *options* *]] [[* *picture* *]]* *[[* **Sum [** INDEX **]** *string* *]]* ... *[[* **Columns** *[[* **[** NROWS *[[* **,** NCOLUMNS *]]* **]** *]] {* *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. Text style and format are applied in the record-wide texts (defined in the next format) and they are inherited to the columns does not contain own definitions. 1 <= INDEX <= 60. **Sum** defines an expression, which is evaluated and collected, when the corresponding record is written out. It is allowed only in a **Level**, a **Total**, or a **Headline**. The definition string contains an *expression*. **Columns** defines the fields of a whole record. If NROWS and NCOLUMNS are not defined, then it is a simple record with arbitrary number of fields. If we define NROWS and NCOLUMNS, then the record is tiled, and the number of

		fields must be NROWS*NCOLUMNS. NROWS >= 1 and NCOLUMNS >= 1. The default value of NCOLUMNS is 1. The default frame of a column of a simple record is the frame of the entire record, of a tiled record is the frame of the entire record equally divided to NROWS by NCOLUMNS grid. The fields are coming row-by-row. Graphic items is a list of simple geometric items, like straight lines, rectangles, rounded corner rectangles, circles and arcs. Simple frames can be defined as border of a field (see *border*).
column	::=	{ *[[frame]] [[textStyle]] [[format]] [[border]]* } This item defines a visible frame in the list. This is an optional field in any record. Its frame is relative to the frame of the record defined in levelTotalDesc. If both left and right values are zero, column does not appear. If both top and bottom values are zero then the item appears at the top of the frame of the record defined in levelTotalDesc. If top and bottom values are equal, but positive, then the item appears with the offset from the top of the frame of the record defined in levelTotalDesc. If left or right is negative, is it relative to the right of the frame of the record defined in levelTotalDesc. If top or bottom is negative, is it relative to the bottom of the frame of the record defined in levelTotalDesc. The default frame of a column of a simple record is the frame of the entire record, of a tiled record is the frame of the entire record equally divided by the number of the rows and the number of the columns. The textStyle and the format inherit the defined values of the whole record defined in levelTotalDesc. The *border* is an entire or a part of the frame drawn by a visible pen.
picture	::=	**Picture** { *[[frame]] [[origin]] [[name]] [[pictureScale]] [[picturePosition]]* } This item defines a picture frame in the list. This is an optional field in any format record. Its frame is relative to the frame of the record defined in levelTotalDesc. If both left and right values are zero, column does not appear. If both top and bottom values are zero then the item appears at the top of the frame of the record defined in levelTotalDesc. If top and bottom values are equal, but positive, then the item appears with the offset from the top of the frame of the record defined in levelTotalDesc. If left or

		right is negative, is it relative to the right of the frame of the record defined in leveITotalDesc. If top or bottom is negative, is it relative to the bottom of the frame of the record defined in leveITotalDesc.
frame	::=	**Frame** *[[* **Fixwidth** *]]* 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. **Fixwidth** can appear in the frame of a *column* only. It has a meaning, if the template is a layout: the field (and all other fields in the column of a tiled record) keep their width, if the number of the columns changes. (Otherwise the total width of the record is divided into equal parts.)
textStyle	::=	**Text** PEN *[[* , *((string* || **Defaultfont** *)) [[* , SIZE *[[* , *style [[* , justification *[[* , truncating *]]]]]]]]]]* This item defines the attributes of the text: pen, font, size, style, justification, and truncating. The default PEN is 1 (1 <= PEN <= 255), corresponding the ArchiCAD pens. The string contains the font name, or ArchiCAD uses the default font of the current operating system. Note! The installed fonts can be different on different systems, so if you do not use the default font, sometimes when you port your template, you should edit the font name. The default SIZE is 9 points (SIZE >= 4). The *style* of the text, the *justification* of the text lines, and the *truncating* of the entire text are also can be set, see at the appropriate places.
style	::=	**Plain** || *((styleItem [[* + *styleItem]]...))* This item describes the style of the text: it can be **Plain** or a combination of other attributes, like **Bold**, or **Italic**. The attributes can appear in any order, but any of them cannot

		appear twice in the same style. The default style is **Plain**.
styleItem	::=	**Bold || Italic || Underline || Outline || Shadow || Condensed || Extended** These are the possible attributes of a text *style*.
justification	::=	**Left || Right || Centered** This item describes the justification of the text lines: each line can be justified at the left border of its frame, at the right border of its frame, or aligned to the vertical centerline of its frame. The default is the **Left**.
truncating	::=	*truncateItem [[+ truncateItem]]* This item describes the truncating of a (multiline) text. If no truncating option is present, then the text will appear as is (no truncating occurs at the border of the frame of the text). If only the **Multiline** option is present, then the text is autowrapped: the long lines will be broken at the border of the text frame. If only the **Truncated** option is present, then the entire text will be cropped at the first long line (and three dots will be inserted). If both **Truncated** and **Multiline** options are present, the each long line will be cropped (and three dots will be inserted).
truncateItem	::=	**Truncated || Multiline** These are the possible options of a text *truncating*.
format	::=	**Format** *string* ArchiCAD will write out this text onto the list. The text will be written with the defined *style* and into the defined *frame*. The default is "" (empty string). The string may contain special format items, these items will be replaced the appropriate values of the processed data.
options	::=	**Options** *((None || ((optionItem [[, optionItem))))* This item describes the options of a record. The default is **None**. The **Pagebreak** option commands ArchiCAD to use a new page **Before** and/or **After** written out the current record. **Pagebreak** is not allowed in a *headerItem*, in a *footerItem*, or in a *backgroundItem*. The **Only** option means: the record belongs to a group of consecutive records (levels/headlines/totals). All of them has the same option, but the record before the group and after the group has not, or they are not existing. Only one record will be written out of the group, if a parameter changes: the record belongs to that parameter. Usually the members of a group are

		the same. **Only** is not allowed in a *headerItem*, in a *footerItem*, or in a *backgroundItem*.
optionItem	::=	*((* **Pagebreak** *[[positionItem [[+* *positionItem]]]]))* || **Only** These are the possible *options* of a record.
positionItem	::=	**Before** || **After** The parameters of the **Pagebreak** option. The default is the **Before**.
origin	::=	**Origin** *((* **None** || **File** || **Preview** || **Data** *))* This item describes the origin of a picture or drawing. The default is **None**. ArchiCAD draws an empty frame with an X (**None**), it uses a PICT (GIFF,...) file (**File**), it uses the preview of an ArchiCAD library part (**Preview**), or it uses the drawings coming from the processed data (**Data**).
name	::=	**Name** *string* This item defines the name of the PICT (GIFF,...) file (**File**), or the name of the preview of an ArchiCAD library part (**Preview**). The default is " " (empty string), but probably you have not a file with this name!
pictureScale	::=	*((* **Scale** || **Drawingscale** *)) ((* **Auto** || *((* SCALE *[[* **,** SCALE *]]... [[* **,** **Auto** *]]))* This item defines the scale factors of the picture. ArchiCAD tries the scales from the first one, until the picture fits into its *frame*. If none of the scales are good, then ArchiCAD selects the smallest one. In the case of **Scale** the numeric value means the real scalefactor (0.02 means 1:200), in the case of **Drawingscale** it means the dividend (200 means 1:200). Note: in the previous versions of list templates the numeric value meant the ratio to 1:100 (0.5 means 1:200). SCALE > 0. **Auto** means the optimal scale for the given *frame*. The default is **Auto**.
picturePosition	::=	**Position** *((* **Lefttop** || **Top** || **Righttop** || **Left** || **Centered** || **Right** || **Leftbottom** || **Bottom** || **Rightbottom** *))* This item defines the position of the picture within the frame. The default is **Lefttop**. The named point of the picture will hit the appropriate point of the picture *frame*.
border	::=	**Border** OFFSET *[[* **,** PEN *]] [[* **,** *borderItem [[+* *borderItem]]...]]* This item defines a visible border around the corresponding *frame*. OFFSET is the distance between the original *frame* and the lines to draw. OFFSET >= 0. The default PEN is 1 (1

		<= PEN <= 255), corresponding the ArchiCAD pens. If no *borderItem* defined, then ArchiCAD draws the entire border, otherwise the enumerated sides. The *borderItems* can appear in any order, but any of them cannot appear twice in the same border.
borderItem	::=	**Top** || **Left** || **Bottom** || **Right** These are the possible sides of a *border*.
graphicItem	:=	*rect* || *line* || *circle* || *arc* It defines simple geometric items to draw into a record.
rect	::=	*((* **Rect** LEFT , RIGHT , TOP , BOTTOM *[[,* PEN *]])) || ((***Roundrect** LEFT , RIGHT , TOP , BOTTOM , RADIUS *[[,* PEN *]]))* This is a definition of a rectangle, or a rounded rectangle. 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 negative, then the program uses the sum of the given value and the bottom of the level or total field. The default PEN is 1 (1 <= PEN <= 255), corresponding the ArchiCAD pens.
line	::=	**Line** LEFT , RIGHT , TOP , BOTTOM *[[,* PEN *]]* This is a definition of a straight line. 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 negative, then the program uses the sum of the given value and the bottom of the level or total field. The default PEN is 1 (1 <= PEN <= 255), corresponding the ArchiCAD pens.
circle	::=	**Circle** CENTERX , CENTERY , RADIUS *[[,* PEN *]]* This is a definition of a whole circle. 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 negative, then the program uses the sum of the given value and the bottom of the level or total field. The RADIUS of the circle should not be negative. The default PEN is 1 (1 <= PEN <= 255), corresponding the ArchiCAD pens.
arc	::=	**Arc** CENTERX , CENTERY , RADIUS , ALPHA , BETA *[[,* PEN *]]* This is a definition of an arc. 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 negative, then the program uses the sum of the given value and the bottom of the level or total field. The RADIUS of the arc should not be positive. ALPHA in decimal degrees is the beginning angle of the arc, measured from the positive x-direction (0 <= ALPHA < 360). BETA in decimal degrees is the closing angle of the arc, measured from the positive x-direction (ALPHA <= BETA < ALPHA + 360) The default PEN is 1 (1 <= PEN <= 255), corresponding the ArchiCAD pens.

Items of the format texts
All characters are copied to the output list except the format _items_, which are replaced the appropriate values of the processed data.

index	::=	*[[* **[** *((Current ||* INDEX1 *))]* *[[* **[** *((Current2 ||* INDEX2 *))]]]]]* This is an index expression of a parameter format item. The primary index is the index of the parameter (1 <= INDEX1 <= 60). The default of INDEX1 is 1. **Current** means the order number of level, total, or headline, which contains the index. The secondary index is the index of the multilevel parameters (for example all components, layers of a composite wall, surface etc.) 1 <= INDEX2. The default of INDEX2 is **Current2**. If **For All** condition is active (see _repetitiveItem_) for the record **Current2** iterates from 1 to the number of the subparameters, otherwise it is 1.
index2	::=	**[** INDEX **]** This is an index expression of a **Usertext** format item (1 <= INDEX <= 15).
index3	::=	*[[* **[** INDEX1 **]** *[[* **[** INDEX2 **]** *]]]]* This is an index expression of a **Sumlevel**, **Sumtotal**, **Sumheadline** format item. The primary index is the index of the parameter (1 <= INDEX1 <= 60). The default of INDEX1 is 1. The secondary index is the level of the sum ("subtotal") (1 <= INDEX2 <= 60). The default of INDEX2 is 1.
item	::=	*((#* || *^)) ((identifier || (((* _expression_ *))) [[* **:** WIDTH *[[* **.** PREC *]]]]* This 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 (0 < WIDTH). The default value for the WIDTH is the width of the input data. The PREC is the number of decimal digits, if the parameter is numeric (0 <= PREC < WIDTH). The default value for the PREC is 0. For **Date** and **Time** PREC has special meaning (see below). 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* || **Keycode1** *index* || **Keycode2** *index* || **Keycode3** *index* || **Keycode4** *index* || **Keyname** *index* || **Code** *index* || **Number** *index* || **Name** *index* || **Totalnumber** || **Current** || **Current2** || **Sum** *index* || **Sumlevel** *index3* || **Sumtotal** *index3* || **Sumheadline** *index3* || **Layoutscale** || **Drawingscale** **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*). **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. **Keycode1**, **Keycode2**, **Keycode3**, **Keycode4** are the 1st, 2nd, 3rd, 4th segment of 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) .

		Totalnumber 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** (see *repetitiveItem*) condition is not active for the level, then **Current2** is always 1. **Sum** is the user defined **Sum** value outside of any Level, Total, or Headline definition (see *repetitiveItem*). INDEX2 = 1. **Sumlevel**, **Sumtotal**, and **Sumheadline** is the current value of the INDEX2th **Sum** defined in the INDEX1th Level, Total, or Headline respectively (see *levelTotalDesc*). INDEX2 <= 60. **Layoutscale** is the scale of the entire layout given in the *layoutScaleItem* command or the scale of the floorplan window of ArchiCAD. **Drawingscale** is the selected drawingscale in a record (see *levelTotalDesc* and *pictureScale*). **Notice**: If the format text is in a *headerItem*, in a *footerItem*, or in a *backgroundItem*, then indentifiers referring to a parameter (**Parameter**, **Quantity**, **Unit**, **Value**, **Total**, **Keycode**, **Keycode1**, **Keycode2**, **Keycode3**, **Keycode4**, **Keyname**, **Code**, **Name**) can cause unexpected results.
expression	::=	*a_standard_GDL_numeric_expression* An arithmetic expression using the GDL syntax. As a variable you can use any *identifier*, as well as the global variables of ArchiCAD. The latter variables are loaded at the beginning of the lst, and ArchiCAD keep their values during the listing. If a variable is textual or its index is (indices are) out of bounds, then zero value will be used. There is a new function (CEIL) to get the smallest greater integer above the argument (the ceiling function). For example: If the expression is 5.5*CEIL(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
            Format      "#number[4] pieces"
        }
    }
    Picture {
        Frame   0, 40, 6, 56
        Origin  data
    }
}

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[3]
                        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  100x100          30 kg            140000 Ft           2 pieces
Single  60x80            12 kg            60000 Ft            2 pieces
Single  80x80            16 kg            80000 Ft            1 pieces
Total:                   100 kg           480000 Ft
Place: 2nd Floor
Double  100x100          30 kg            140000 Ft           1 pieces
Single  60x80            12 kg            60000 Ft            2 pieces
Single  80x80            16 kg            80000 Ft            2 pieces
Total:                   86 kg            420000 Ft
Grand Total:             186 kg           900000 Ft
```