# AreaList Pro

## Version 11.4 .1 Release notes

### February 17, 2024

Published by **Plugin Masters** worldwide

### System requirements

AreaList Pro version 11 is compatible with 4D v18 and over for MacOS (10.14 or over, Intel and Silicon, signed and notarized) and Windows (64-bit mode).

This version also supports v8 API and v9-10 API: your existing code should not require any changes.

### Version 11.4.1

- Enhanced Excel export
  - column width matches width in ALP (there is a new call in LibXL API to set pixel width)
  - fixed a bug in export of attributed (styled) text when the text contains no formatting (the text was exported as is, without processing XML entities)
- Fixed a bug in mapping cell value to 4D value type (could make problems with Collections when column type is set to implicit)
- Changed click area computation for popup
  - you don't have to hit that small icon, full cell height is used for the popup-click detection
- Fixed *AL_SuperReport* when calculated columns are used
  - now the template contains a proper pointer to the internal ALP array which should be already filled with values
- Added support for ALP footer row when printing
  - PLP didn't support footer row, now it will properly print it
  - note that when using breaks and break footer for level 0 is defined, ALP displays that in the footer
  - when ALP is to be printed and it shows real ALP footer (and not the Total break footer), it is added as break footer to PLP
- Changed handling of ALP_Column_DisplayControl - it is now ignored for break rows (breaks use only text values, to format them as e.g. checkbox is nonsense)

- Fixed printing of rows spanning multiple pages
  - previously integer/real columns using default horizontal alignment (mapped to right alignment) were reprinted on every continuing page, now they are printed only at beginning
- Fixed printing when ALP_Area_LimitRows is set (was not honored)

## Version 11.4 – New

- Added missing constants for enterability
  - AL Column entry popup no meta (4)
  - AL Column entry both no meta (5)
  - AL Column entry popup cellwide (6)
  - AL Column entry popup cell NM (7)

  - AL Cell entry typed only (1)
  - AL Cell entry popup only (2)
  - AL Cell entry both (3)
  - AL Cell entry popup no meta (4)
  - AL Cell entry both no meta (5)
  - AL Cell entry popup cellwide (6)
  - AL Cell entry popup cellwide NM (7)
- Picture scaling on Windows using scaling other than 100%

## Version 11.4 – Changes

- Changed maximum allowed value for ALP_Column_Enterable and ALP_Cell_Enterable, maximum is now 7 – when enterability is 7, click anywhere into the cell is handled the same as click on the popup icon (identical to using value 6), but when building the popup from arrays the meta characters are ignored
- Refactored C_POINTER usage – due to changes in 4D v20, all usage of C_POINTER variables was refactored to not use "persistent" pointers (e.g. pointer to array has no reason to be modified, but 4D v20 does exactly that and ALP did not expect it could change)
- Changed implementation for entry click (when set to single-click without modifiers) and entry is disabled (cell not enterable) — now it will not immediately report a click, should be able to handle drag (and report a click when D&D is not started)

## Version 11.4 – Fixes

- Fixed drawing of popup icon when default vertical alignment is used and the row displays multiple text lines: intention was to draw it in middle of first line, not in middle of the whole row

- Fixed handling of a click onto a popup icon: the code was not adjusted for the changed position used when drawing, now it will match the rectangle used when drawing

- Fixed handling of ALP_Area_EntryAllowTabIntoPopup when a cell is skipped using ALP_Area_EntrySkip in a callback: the cell was enterable by typing even when configured to be enterable by popup only

- Fixed text drawing when a single line has to be drawn and ALP_Area_NumRowLines is not 1: at some conditions (bad text measurement) nothing was drawn

- Fixed text measurement on Mac: some fonts use leading which was not used in text height computation and due to previous point nothing was drawn

- Fixed background color drawing in "empty column" on the right side when both alternate row color is to be used and row has defined background color: used was alternate color, not the row color is used

- Fixed a crash with RESOLVE POINTER and 4D v20 beta (4D is modifying input-only parameter $1!) — could be there are other places (other commands) which have such bug in newest 4D

- Fixed break header processing when the only break variable used is \BREAKVALUE (bug new to ALP 11.3b3)

- Fixed entry widget rectangle computation/redrawing
  when cell background is set (e.g. to green), some artifacts can be visible at end of the line (after half a second - when widget is redrawn to remove caret (to make it blink))

- Fixed printing multiple rows with PRINT SELECTION (bug new to ALP 11.0) — depending on usage it could work, fail to print anything after first record or miss breaks

- Fix for decimal alignment when format uses "##0.###"

- TextFormatter fix for bad caching (non-left aligned text must be recomputed) -> SUM in last (expanded) column is properly positioned — this also fixes bad row height computation when printing (PLP) and not using breaks

- Fix for decimal alignment
  ALP_Column_HorAlign/ALP_Column_FtrHorAlign/ALP_Cell_HorAlign is set to AL Just decimal (6) and the cell is displaying real number, when format uses multiple "#" at end (no trailing zeroes), e.g. "##0.###"

Windows specific:

- Fixed entry rectangle size on Windows when Windows Scaling is > 100%

- Changed some code related to "high DPI support", the entry rectangle/font size is now correct for DRP to Windows 2011 server

- A "leading ghost caret" during entry should be fixed (was always visible when ALP_Area_WindowsText = 1 (GDI))

- Fixed rectangle frame drawing when ALP_Area_WindowsText = 1 (GDI) (bug new to ALP 11.3b5) - missing focus, difference when drawing D&D drop target line

<p style="text-align:center"><span style="color:magenta">**Version 11.3 – New**</span></p>

- • Added a new property for scrollbar behavior
  - name ALP_Area_ScrollbarStyle
  - value "scst"
  - type Boolean
  - read/write/persistent
  - xml name "scrollbarStyle"
  - default 0
  - -> 1 means "use overlay mode" - visible only while scrolling (with mouse wheel), scrollbar does not consume space in the area, it is drawn temporarily over the data
- Added persistency for property ALP_Area_ScrollbarArrowsPos, xml name "scrollbarArrows"
- Added property for limiting long text
  - name ALP_Area_MaxRowLines
  - value "rowm"
  - type integer
  - read/write/persistent
  - xml name "maxRowLines"
  - default 0, min 0, max 64

When ALP_Area_NumRowLines = 0 (variable row height) and this property is set to a value greater than 0, at most ALP_Area_MaxRowLines will be displayed

In that case when ellipsis is to be used (e.g. ALP_Column_UseEllipsis), last line will display it

NOTE that ellipsis addition to last line is implemented only for left aligned text (default alignment for text is mapped to left), does not work for any other alignment (where ellipsis is at the beginning or in the middle of a long line)

ALP_Area_MaxRowLines differs from ALP_Area_MaxCellHeight in:
  - number of lines of text instead of points
  - ellipsis for left aligned text can be drawn at end of last visible line

Example: having [left aligned text] "111\r222\r333" and ellipsis active (e.g. ALP_Column_UseEllipsis):
ALP_Area_NumRowLines = 1 (old AreaList Pro v7 behavior: ignore \r, draw single line)
111222333
ALP_Area_NumRowLines = 0, ALP_Area_MaxRowLines = 0:
111
222
333
ALP_Area_NumRowLines = 0, ALP_Area_MaxRowLines = 1:
111…
ALP_Area_NumRowLines = 0, ALP_Area_MaxRowLines = 2:
111
222…

ALP_Area_NumRowLines = 0, ALP_Area_MaxRowLines >= 3:
```
111
222
333
```

- Added property for getting current system highlight color
  - name ALP_Area_HighlightColor
  - value "uihc"
  - type integer
  - read/write
- ALP_Area_HighlightColor is writable, this color will be used in place of the system highlight color
  - to reset it to use the system color again, set the value to 0
  - Note: this is a global property, it will affect all areas
- Using native appearance (ALP_Area_Appearance = -1) on Mac finally supports accent color for checkbox/radio button
- Added a new command to get a picture as used internally in AreaList Pro
  - *AL_GetIcon* (areaRef:L; iconID:L) -> Picture
  - current appearance of the area is used to resolve special icon IDs (areaRef can be zero)
  - can be used e.g. to get native appearance pictures (ALP_Area_Appearance = -1) and use them with pictures appearance (ALP_Area_Appearance = 11)
- Added constants for miscellaneous internal icons (popups, node disclosure triangle, checkbox, radio button, sort indicator)
  - AL Icon Generic Popup PictID
  - AL Icon Date PictID
  - AL Icon Time PictID
  - AL Icon Node Right PictID
  - AL Icon Node Down PictID
  - AL Icon CB Unchecked PictID
  - AL Icon CB Checked PictID
  - AL Icon CB Mixed PictID
  - AL Icon RB Unchecked PictID
  - AL Icon RB Checked PictID
  - AL Icon Sort Up PictID
  - AL Icon Sort Down PictID
  - AL Icon Appearance Offset

When using *AL_GetIcon*, current ALP_Area_Appearance is used to resolve the real picture ID, but when setting the icon, nothing special is done, you must use AL Icon Appearance Offset to modify icons for ALP_Area_Appearance = 11 (AL Use Pictures Interface).

<center>**Version 11.3 – Changes**</center>

- Added support for Win11 native icons (header, checkbox, radio button, scrollbars)
- Added new appearance constant for Win11 - name <u>AL Force Win11 Interface</u>, value 12
- Changed maximum allowed value for <u>ALP_Area_Appearance</u> from 11 to 12
- Header drawing changes:
  - fixed header drawing
  - -> no divider separator above the vertical scrollbar
  - added possibility to not draw the divider separator for last column (on the right)
  - name <u>ALP_Area_HideLastHdrDivider</u>, value "hhld", type Boolean, read/write/persistent, xml name "`hideLastHdrDiv`", default 0 (False)
- Changed drawing of headers, especially on Windows (try using <u>ALP_Area_WindowsText</u> > 0)
- Changed position of popup icon computation (it is not always at top, depends on line height and vertical alignment)

<center>**Version 11.3 – Fixes**</center>

- Fixed a possible crash in old API (e.g. using <u>AL_IsValidArea</u> with area 0)
- Fixed slow drawing ("row by row") after hierarchy is sorted in Fields mode
- Fixed handling of arrows in scrollbar: when using **GOTO OBJECT** in a form and then displaying a new form with AreaList Pro, it receives dozens of IDLE events - too fast scrolling
- Fixed entry mode on Windows when Windows scaling is used (with 4D v19 Rx; Windows scaling support implemented in 19R4)
- Fixed possible freeze parsing malformed styled text
- Fixed endRow for "\\FUNCTION" callback in break header when "\\function" is the only "break value" used in the break header cells
- Fixed printing of repeating break values when configured to be omitted (not drawn)
  - -> same behavior as old PrintList Pro
- Fixed column sizing when <u>ALP_Area_ResizeDuring</u> is in effect and data are updated
- Fixed bad header drawing after a column resize as effect of a double-click on header column divider

<center>**Version 11.2.2**</center>

- Fixed printing of breaks

## Version 11.2.1

- Fixed a crash when invisible rows are used and user clicks into empty area below data (bug new to ALP 11.2)
- Fixed handling of invisible columns (would fail for EntitySelection) for D&D and Copy


## Version 11.2 – New

- Added missing constant and accessor for default time format (which defaults to "0")
  - name ALP_Area_DefFmtTime, value "dfti"
  - when adding a column using time field or array time, this format is used as default (added for 4D v14 ARRAY TIME back in 2013)
- Added ability to export only defined columns:
  - set the list of columns to export (using ALP_Area_ExportList or ALP_Object_ExportList)
  - when setting ALP_Area_ExportOptions, include the AL Export Using List option to use the list of columns
- Added constants for that:
  - added area property ALP_Area_ExportList, value "expL", type text, read/write/persistent, xml name "exportList", default ""
    - use comma separated list of column numbers, e.g. "11,3,4,9,1" -> export only these columns in this order
    - **Note**: when moving columns (by D&D (and ALP_Area_Compatibility = 1 or ALP_Area_DragMoveColumns = 1) or with *AL_MoveColumn*), this list is automatically adjusted (column numbers are renumbered according to the move)
  - added constant ALP_Object_ExportList (value "expL") to be used with *AL_GetObjects/AL_SetObjects*
    - expected is ARRAY INTEGER or ARRAY LONGINT containing the column numbers
  - added constant AL Export Using List (value 65536) to be used in ALP_Area_ExportOptions property
    - when this value is used and ALP_Area_ExportList is not empty, only columns defined in ALP_Area_ExportList will be exported with ALP_Area_ExportToFile
  - changed maximum value allowed for ALP_Area_ExportOptions: max = 131071 (0x0001FFFF)
- Added new property to allow sorting selection by calculated column in Fields mode
  - name ALP_Column_SortFormula
  - svalue "cobf"
  - type text
  - read/write/persistent
  - XML name "sortFormula"

- o default ""
- o Set this property to a method name which will be called by ORDER BY FORMULA as *Method* ($alp:int; $colNum:int) -> value to use for order
- o Note that $colNum is negative for descending order - ORDER BY FORMULA explicitly uses descending order, you should not revert it
- o To be usable by end user (click on header or Sort Editor), you must explicitly set ALP_Column_Sortable to 1 (calculated columns are not sortable by default).

  e.g.
  ```
  $err:=AL_AddCalculatedColumn($area; $colType; "MyColumnCallback")
  AL_SetColumnTextProperty($area; $col; ALP_Column_SortFormula;
  "MyColumnSortCallback")
  AL_SetColumnLongProperty($area; $col; ALP_Column_Sortable; 1)

  // MyColumnSortCallback
  C_LONGINT($1)  // ALP
  C_LONGINT($2)  // column number, negative when descending order
  will be used (using explicit "; <")
  // C_xxxxx($0)
  $0:=MyCalculatedValueForCurrentRow.call($1; Abs ($2))
  ```

## Version 11.2 – Changes

- Added support for hidden columns used in hierarchy
  - o first visible [data, not the one defining the first hierarchy level] column properly displays data when it ends as first column even when using auto-hierarchy

- Added support for auto-hierarchy without creating the break headers
  - o when first column in the list set with ALP_Area_AutoHierarchy/ALP_Object_AutoHierarchy is negative, AreaList Pro does not change the break headers, you can use custom headers (e.g. with \SUM in any data column)

- Format is now resolved recursively (up to 5 times)

- Export (Excel) does not use format when it is not resolved (starts with '|' or ':')

- Added auto-width computation for break rows, including total displayed in footer row

- Relaxed *AL_DuplicateColumn* requirements:
  - o old and new data pointer can use different data type
  - o developer is responsible for properly re-setting relevant properties (e.g. ALP_Column_Format)

- Added implementation for formats when using 4D in project mode (by parsing the main DB's filters.json file)
  - o formats should be properly resolved (needed for Excel export)

- Changed Excel export to set Wrap mode implicitly when ALP_Area_NumRowLines is not 1
  - o Excel will not wrap when text contains "\r" unless wrap mode is enabled

<h1 style="color:magenta; text-align:center">Version 11.2 – Fixes</h1>

- Fixed "syntax error: collection or object expected" alert when retrieving field type from "This.relatedEntity.anotherRelatedEntity.entityName"

- Fixed various issues related to breaks

- Fixed drawing of column separators in break rows

- Fixed selection with up/down arrow keys (esp. when break processing with hierarchy is used)

- Fixed possible crash when changing selection (and clearing data cache) while break processing is used

- Fixed auto-width computation for hierarchy when [leading] break columns are invisible

- Fixed multiple-row selection mode (anchor row not properly set with drag-select, next keyboard navigation (using shift key) selects incorrect row)

- Fixed an old bug related to 4D v11+ when "use comma and period as placeholders" is in effect
  - when it is active, default column format must not be using localized version of the format, but the US format (e.g. even for French locale use "#,###,##0.00", not "# ### ##0,00")
  - -> non-localized format is used as default column format for integer, longint and real (set at startup)

- Fixed crash (memory corruption) when using *AL_SetIcon* to replace existing picture (bug from ALP 11.0b10)

- Fixed binary Advanced Properties loading in Design (didn't include any column) (bug new in ALP 11.0b11)

- The area properly redraws when setting hierarchy expansion state programmatically

- Fixed implementation for ALP_Area_PrintArea
  - it was failing to properly initialize the printing area when arrays mode was used with arrays in different scope (ALP in main DB but pointers to arrays from a component or vice versa)

- Fixed drawing when many rows are invisible in Fields mode (by not fetching invisible rows from records)
  similar (identical, but in different part) problem was fixed in 10.3b2 - ALP in Arrays mode

- Fixed drawing of headers

- Fixed bad scrolling during entry (old ALP 10.1 bug, at specific conditions, an area could scroll the entry cell out of view on every key down)

<h1 style="color:magenta; text-align:center">Version 11.2 – Windows specific</h1>

- Modified property ALP_Area_WindowsText from type Boolean to type integer, min = 0, max = 2, default = 0
  - 0 - GDIPlus

- o 1 - GDI
- o 2 - DirectDraw + DirectWrite
- Changed handling of pictures (resolution/DPI, new implementation for conversion for GDI & DD)
  - o -> could be the icons/pictures are too small (there was no conversion for 72 dpi vs. 96 dpi, now there is one)
- Fixed Windows scaling (4D v19R4+)
  - o -> click into area, entry widget position, text size, pictures, …
  - o scrollbars and headers are drawn differently to get rid of white hair lines between elements

<div align="center">

### Version 11.1

</div>

- Fixed type-ahead
- Fixed <u>ALP_Area_Name</u> getter
- Fixed header "disappearing"
  - o under some specific conditions, cells in header were not drawn
- Setter for <u>ALP_Area_TypeAheadString</u> now executes the type-ahead in focused target when possible - similarly to user typing the specified text
  - o now, you can scroll/select row in AreaList Pro with code, e.g. assuming Example 15 is sorted by city, you can select a row:

```
AL_SetAreaLongProperty(eList; ALP Area SortList; 1)
AL_SetAreaTextProperty(eList; ALP Area TypeAheadString; "Orlando")
```

- Added properties to <u>ALP_Area_PrintOptions</u> object for standalone printing
  - o "drawFrame": int, 0-2 — when present and in range, <u>ALP_Area_DrawFrame</u> is set to this value, otherwise <u>ALP_Area_DrawFrame</u> is set to 1 when it is 2
  - o "printHeaders": int, 0-2 — when present and in range, <u>PLP_PrintHeaders</u> is set to this value, otherwise <u>PLP_PrintHeaders</u> is set to 0 for <u>ALP_Area_HideHeaders</u>=1 and 2 for ALP_Area_HideHeaders=0
  - o "frameWidth": real, 0.0-1.0 — when present and in range, <u>PLP_FrameWidth</u> is set to this value (nothing is done otherwise)
- <u>PLP_FrameWidth</u> default is 1.0 and is not modified when printing
  - o to be used, <u>ALP_Area_DrawFrame</u> must be 1
- Changed maximum allowed value for <u>ALP_Column_Enterable</u> and <u>ALP_Cell_Enterable</u>, maximum is now 6
  - o When cell enterability is 6, click anywhere into the cell is handled the same as click on the popup icon
- Fixed popup icon click detection - possibly wide column separator is now honored

<h1 style="text-align:center;">Version 11 – Major new Features</h1>

- **Sub-ALP** is the most fantastic new feature, that you won't find anywhere else. A column displays cells that really are AreaList Pro areas with their own formatting, like invoices for a customer, invoice lines for an invoice, etc.

- **Collections** are now supported for data source.

- **Entity Selections**: now you can integrate 4D's ORDA functionality into AreaList Pro! Use an Entity Selection in your AreaList Pro area.

- **PrintList Pro** is now included as a license option: only one plugin instead of two (no change of PrintList Pro code needed).

- **Standalone Printing**: in addition to PrintList Pro or SuperReport Pro printing of an AreaList Pro area, a single new command just prints the area with all formatting.

- **Export to Excel** (XLS or XLSX) including formatting, with or without header, selected rows or whole area, and many more options (including Sub-ALP).

- **Offscreen** AreaList Pro areas: build an area and Print it or export to Excel without displaying it on screen, with a single command.

- **Apple Silicon** (M1, M2) support.

- By popular demand, the return of the **PDF Manual**, fully updated, including PrintList Pro and detailed explanations of v11 new features.

See the [manual](#) and the [forum](#) for additional details and Hints & tips.

<h1 style="text-align:center;">Upgrading from previous versions</h1>

Any v11 license key will activate any v11.x version.

A specific **Print option** license is available to enable printing with the PrintList Pro module or Standalone. Old PrintList Pro plugin must be removed from the Plugins folder.

Upgrades from versions 9.x and 10.x (single user or server) are available at [pluginmasters.net/alp](#).

<h1 style="text-align:center;">Contact</h1>

Upgrade requests, sales and license information: [orders@pluginmasters.net](mailto:orders@pluginmasters.net)

Technical issues, discussions and requests: [forums.e-node.net](#)

AreaList Pro web page: [pluginmasters.net/alp](#)