



Knowledge Inside
arKItect your system

arKItect 6.0 Release Notes

arKItect 6.0 provides major improvements:

- Multi-language: a french version is now available but translation in any language can be integrated in a few days,
- Python 3 API is now fully operational and substitutes to Python 2 API. Qt version has changed and it's now possible to integrate seamlessly new browsers and many incredible Py3 libraries,
- Support for 2D CAD design and interface to OpenStreetMap are now available and many improvements have been made since early introduction in arKItect 5.1
- Many User X improvements in order to improve efficiency in using arKItect and all the solutions based on it,

Version is distributed in 64 bits but can be made available in 32 bits upon customer request.

Multi-language interface	2
Techno – shift capabilities	3
Hints– continuous improvement.....	4
Bug fixes	8



Multi-language interface

It is now possible to get arKItekt in any language with no or low effort.
At the moment English and French versions are available:



Take care it is also needed to translate your data model and messages in your scripts but we have developed tools to help you doing that quickly. If you need translation of your data model or translation of arKItekt in another language, please contact us.

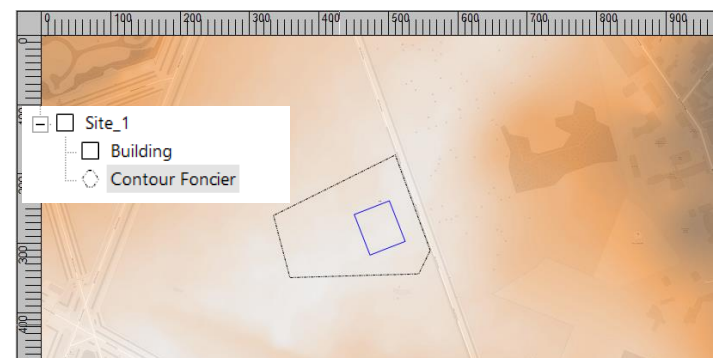
OpenStreetMap interface

We now have python interface in arKItekt allowing to open a browser with OpenStreetMap, edit objects that you can then import in arKItekt and conversely. This feature is available for 2D views as it has to be compliant with a scale. Here is an example where the map in arKItekt is imported and integrated with altimetry analysis. Light zones are lower than dark zones here.

We can provide libraries for computing altimetry of every surface or point (e.g. on a network) in the map. This is one of the many possible applications. In this example, the polygon outline is the site outline and the rectangle outline is a building.

Changes in the platform are also implemented to support GIS interface:

- add default GEO point into Global properties
- save geo points information





Knowledge Inside
arKItect your system

Techno – shift capabilities



Python 3 API

Python API has been a core function of the arKItect platform from the very beginning. Scripts on events mechanism is a key feature to develop powerful real time applications.

arKItect 6.0 python API is now fully Python 3 compatible and ref languages for scripts is Python 3.11.9. If you use PyQt, reference version is now PyQt 5.11.

For legacy scripts translation from py2 to py3, you can update scripts with - py2to3 (package in the standard python installation). However, this will require tests and fixes e.g. if py2 operation / is returning int, in py3 after translation it returns float, to return int you need to use //.

The approach consists in applying known changes first of all (new standard packages, new structures, obsolete methods, etc.) so that you don't need to replace print "A" with print("A") manually. In the transition process, there is option s that lists changes only, makes backups and so on. But in the end, you need to test and tune the code manually. However, should not be a big deal eventually as the transition to Py3 is rather easy.

If you use PyQT, you may be interested by packages like PyQt4toPyQt5. Use of PyQt in arKItect scripts is usually limited so you should be able to do it by yourself quickly, for big applications, the package may be needed.

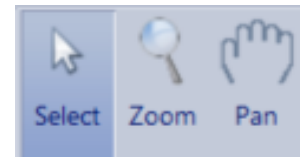


Hints– continuous improvement

Improved zoom management

This feature is very important especially in complex views with thousands of objects in order to focus to a small part of the diagram and come back to the landscape quickly.

At the moment, there are three main modes in any arKItekt view, the select mode allows editing the models including moving or resizing objects in a view, the Zoom mode allows focusing on a part of a diagram in a viewpoint. The Pan mode is used for read only navigation.

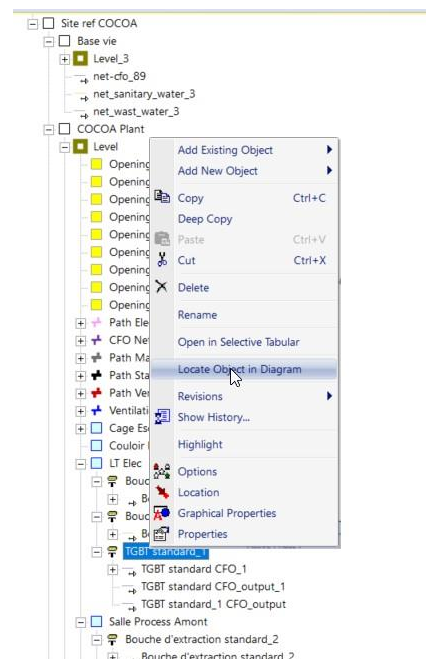


On diagrams with hundreds or thousands of objects, you usually need to switch very often from Zoom to Select mode and each time your mouse is going from the diagram to the mode button and comes back. This can be cumbersome. We decided to include this kind of switch as shortcuts with the keyboard, so that you can Zoom in, Zoom out, and edit seamlessly. There are two new features:

- Zoom in by rectangle selection with Ctrl (no need to go to Zoom mode). Take care if you do not release mouse before Ctrl, then you will remain in Zoom mode, otherwise you will stay in select mode.
- Zoom to fit by Ctrl + mouse right click (no need to click on special button)

Improved “locate object” in complex views

This is the kind of feature that is absolutely needed when you work on very complex diagrams. Locate feature is available for a long time in diagrams but not directly from objects in the treeview. For customers who manage equipments on a linear infrastructure in a 2D view or the ones who look at a site or a complex building, finding an object quickly on the map is a key feature. Its now very simple from the treeview, with a right click and selection of menu “Locate Object in Diagram”. As a result the Internal Block Diagram is centered on the related object which is highlighted because selected.

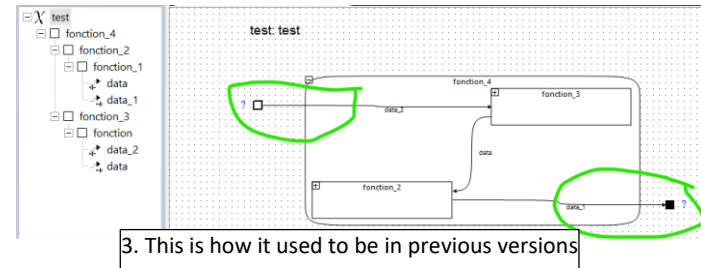
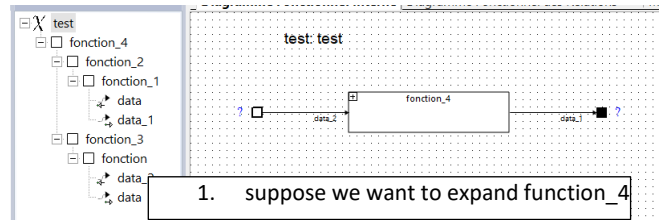
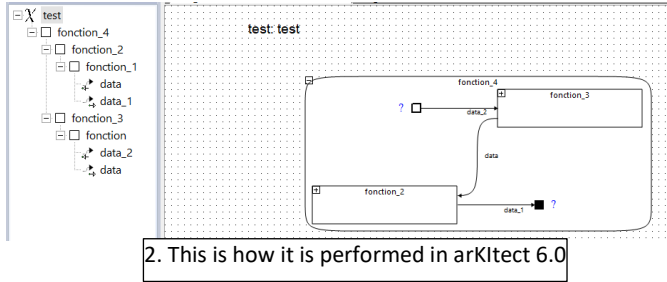


Improved UX

- location: use attribute instead of object name if applicable, while before only the name/ID of objects could be used. So this display depends on active view. However, for objects not displayed in the view, they are displayed by their name.
- redefine text transparency in treeview per type: it’s now possible so see labels or not independently for each viewpoint in arKItekt. Of course, it’s more work in configuration, if you have 10 views and want to display some flow label in 5 and not in the 5 others, you need to configure each view separately.
- allocated icon for object with enum value icon: it is now possible that an object icon inherits the icon of an enum value. Enumerates are often used to limit the size of the data model by defining subcategories of objects. This feature allows limiting the drawback of using an enumerate instead of several types and in the same time you keep advantage of limiting the size of your data model.



- reset outerports: in the past, when expanding a block, outerports would be out of the outline of the block. Now they stay close to producing and consuming sub blocks. The new approach is explained in the use case below:



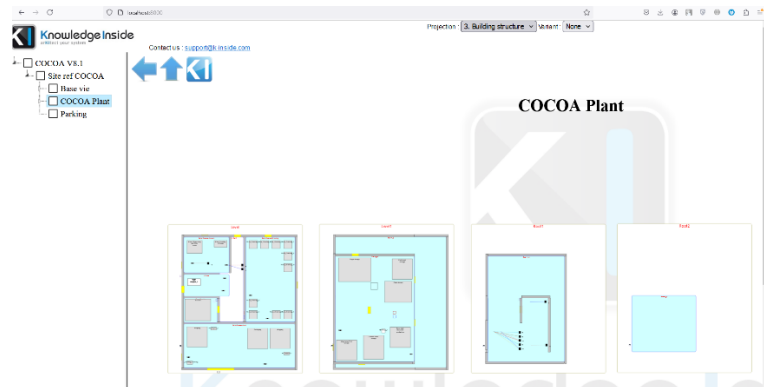
- reset outerports: do not call script anymore (leave it to arKItekt only). We have internalized the automatic display of outerports as this is very often needed by users. In the past, if you would move a block and forget to select it's outerports (interfaces), these outerports would stay in their initial position. Now outerports sticks with their consuming/producing object. This is similar to previous feature.

Performance management

- image manager - release cache on project close, make independent cache for image choose dialog. This is particularly important for 2D features as we can integrate very big images of 10Mo or more so memory management starts to be an issue, even in a single session.
- performance of display of any diagram (gain about 20%) - GetContainerObj
- tuning of views update when changing attributes (set update flag for non-enumerators only if attribute is displayed or used as displayed value for label)
- no intermediate gui updates during some operation (add object, move object)

HTML export of 2D viewpoints

It is also possible now to export 2D maps at scale using the HTML export.
You can also custom the html export with python allowing to tune finely what you really want to display. Please refer to the documentation.



Object shapes, position and size management in 2D views

- for 2D Zone and Polygon shape - define its shape on object creation
- X pos, Y pos as separate update window in properties, this allows avoiding an intermediate properties refresh between both.
- calculate surface of polygons, comes with introduction of polygons shape.
- disable level size modification (use script to synchronize it with building 2D type) Notice Level and Building are 2D types that can be used for other applications. E.g. Building 2D may be physical display of an automotive and Level2D could be specialized in a Door.



- do not show SizePos dialog anymore. We used to display this menu on double click on an object previously.
- initialize objects positions in diagram; fix first position (remove +3 additional units to size). This used to be a problem when adding a 2D object in a treeview without locating it during creation.
- deep copy in 2D diagram should save new position of objects (everything if including position of the source object used to be replicated but this results in having the copied object at same location as the genuine object in a 2D view, ignoring the mouse position where you want to copy duplicated object)
- base move in 2D diagram is 1 unit and not 2
- make possible drag & drop into object graphically while link is created to another object (with lower Z-order; it is about CP without network).
- when rotating create always a group component and rotate it; make track image correspond to rotated components
- special behavior if flipping (due to rotation changes)
- rotated components can be aligned now
- properties - allow editing position and size information together
- Surface label parameter for Level, Room (possibility to hide it)

Improved physical networks management

- Single Port for 2D (mainly makes useless CArkiOnePortCircleComponent). This applies to Crossing Point 2D shape type.
- Split networks: it's possible to delete an edge in a network, in that case we may get two subsequent networks in the case the network has no loop.
- do not allow to change default size for CP in Networks and Level in Buildings. For instance, Level size inherits Building size.
- CP of closed network should be easily selected like CP of normal network. Selection can be made with right click and selection of a zone around the CP to be selected. This process allows to select very small objects even with a big scale in 2D diagrams.
- allow merging any CPs of the same network

Measure

We defined a new category of 2D objects in order to add measures to diagrams. This is often needed when producing deliverables at scale.

- Measure objects: allow move by drag & drop of the link
- add Ruler shape as a new 2D type
- Ruler as a new version of Measure (use of network)
- change arrow symbol for Measure 2D type

More flexible objects scale 2D diagrams

Many features have been implemented to provide freedom to customize 2D views in order to emphasize a particular type of objects. For instance, you may change the size of a network in one specific view for the purpose of producing a specific document.

Here is the list of changes implemented:

- make Font factor float to allow decrease the font size
- border line width vs link line width can be modified in object, type and treeview properties
- refresh components in diagram: take into account line width for selection and line



width factor

- fix display title save for object
- fix Z pos in object; scale factor in treeview
- update views on scale factor change
- Development of new API: add GetScaleFactor, SetScaleFactor for ArkTreeViewObj
- when activating 2D always come back to 100% magnification first (internal bug)

Objects overlap, display and select capabilities and priorities

Some changes had to be implemented in 2D views w.r.t. rules in logical views. Especially related to the overlap and display priority between objects.

Several changes implemented for this:

- allow objects overlap in 2D diagrams
- allow superposition of other objects with CP (CP are topmost objects so ok)
- allow select link label when CP is nearby (CP has still got priority over link itself)
- drag&drop using small cross now follows usual arki drag and drop routine

Performance for 2D

Performance issue may happen with models of several thousands of objects. Especially because maps are flat displays, so we don't benefit of hierarchical definition. So we all the time try to import response times and avoid unnecessary refresh.

- display 2D diagram (skip analysis of xml for links) (gain about 30%)
- cp insert (on edge or by double click) will result in local refresh (only if no script is launched)

Miscellaneous for 2D diagrams

Eventually some changes have been performed that do not fit in previous categories:

- Go to Projection from Location - stop on last possible own diagram (e.g. for a CP do not activate Network diagram, instead stay in upper diagram - Level or Room)
- API: make ARK_TYPE_LABELPOS, ARK_TYPE_DDTYPE, and ARK_TYPE_MUTYPE int constants and not string constants
- make it possible that CP can exist without networks and behave the same way. This allows using this shape properties and priority for other objects that networks edges.
- allow displayed attribute to be used as main additional label
- make DDTYPE_COMMENTS use non 2D label
- DDTYPE_GROUP as 2D type (not displayed used to group objects inside)
- when refreshing labels - refresh new labels first
- add Comments as new 2D types
- allow starting linking objects with only one port using Space+MouseLeftClick

Miscellaneous not related to 2D diagrams

- introduce CObjectComponentWithImage and make it base class and not CObjectComponent
- save ports information only if set and modified, so if there is not change no save and this means automatic display is applied but also it optimizes behavior.



Bug fixes

As usual, we inform you about all fixes performed with this new version.

Fixes related to arKItect 5.1 except for 2D related features.

- crash arKItect when trying to execute script for ArkObj instead of ArkObjRef
- attribute is displayed name (changes in properties, refresh of diagrams)
- crash when editing rtf attribute in Properties with Search dialog left open
- on refresh never set focus on treeview of the displayed model (delete is called twice)
- scroll and image (BaseModel) sometimes objects are not in their real position - redrawn in another location after hovering them with mouse
- link shape when it is bottom to bottom / top to top / right to right / left to left
- when adding objects into expanded object through treeview, objects are placed in the center on top of each other
- don't refresh properties when action (add object) is not finished (it may cause problem when events are used due to Markup which itself uses scripts for rendering)
- calculate additional update area for rotate and move symbols only if component is selected
- HighlightObjects crashes arKItect if ArkObj is passed in parameters
- BBox: drag & drop flow from treeview, dict, location, search
- problem with links in non 2D diagrams when link was shown with arrows coming from inside
- add existing property ARK_ATYPE_TIME_MODIF into constants and allow its using in get method
- API: remove restriction for OnChange event in GetAttributeforEvent (keep it in Set.. method)
- use of CArkImageComponent (inherited from CODImageComponent) image component instead of direct use of GDI plus (fix problem with bad image quality) - necessary regression - no clip image into polygon - to be done using C#

Here are fixes related to arKItect 2D related features.

- when Restore object in Graphical properties is checked - always keep 2D properties like position, geometry
- hide cps in network under variant/phase if network is not displayed
- Text Transparent by type in treeview
- surface change of unit, in square meters/millimeters
- z pos is now in meters only
- precision for position is 4 points
- allow merge cp for Network_closed
- properties - allow editing position and size information together
- after moving object in 2D update Properties
- save xml - fix in expanded property
- do not use partial info when drawing links - only full match, otherwise default ports
- fix z pos save
- important: use of GeBaseRgn(). GetBounds() instead of GetBounds() for new components with labels - label maybe bigger than object and it should not interfere into calculation
- crash when creating new link (due to link line width factor)