

Composition of an Excel File

We will first have a quick look at all the possible sheet types and then we will have a closer look at each of them.

There are three types of Sheets available in an Excel file

Default	Schedules	AHU	~Structure	~Address Help	~Model	~Variables	~Reminder
---------	-----------	-----	------------	---------------	--------	------------	-----------

The Data Sheets

Those sheets contains all the components you wish to import/export. For a sheet to be considered as a data sheet, the sheet's first cell (A1) must be the internal id column and the sheet can't have the same name as one of the five standard helper sheets. Example: above "Default", "Schedules" and "AHU" are data sheets.

	A
1	Internal id
	Id to reuse the

The Standard Helper Sheets

There are five possible helper sheets in the excel template files:

- **~Default Sheet:** This sheet is used to create a tab during an export if this tab is missing. This sheet doesn't contain any data, only a header
- **~Address Help:** This sheet shows you how to fill the Address Column depending on the component
- **~Model:** This sheet lists the Active-Framework nodes by aspect.
- **~Structure:** This sheet will determine how many data sheets there will be, how they will be named and which components will be included inside
- **~Variables:** This sheet will allow you to store variables that you will be able to reuse later (like some shortcuts)



Since the 46.16.0, the ~Help sheet was renamed to ~Address Help for more clarity. If you updated your btibExcel from an older version, please also update the config files (use the "Update default config" button in the import or export tool, see the tools in the [Main concepts](#) for more informations). The ~Help sheet will be automatically updated in your exports

Your Helper Sheets

You can also add your own helper sheets with any information you want. But if the first cell is named Internal Id, the sheet will be considered as a Data Sheet.

By convention, we advise you to add a prefix ~ to its name. Example above ~Reminder

The ~Default Sheet

Internal id	Base	Template	Name
Id to reuse the component. Use#	Reference where the component will be created	Template name with or without subpath	Name of the instance to create

In the default config Excel files, you will always see a 'Default' sheet. It's the one by default used as a template to create a Data Sheet in your excel file during an export. See the EXPORT part below in the description of the ~Structure Sheet for more information.

The ~Address Help Sheet

This sheet shows you how to fill the Address columns for different types of communicating components. If the component you're looking for doesn't appear in the list, it means that it's not supported yet. Contact us in this case.

The ~Model Sheet

In the Excel File Templates, the sheet will be empty. This sheet will be filled after an export if you use Active-Framework. Each column will be associated to an aspect and they will be filled with all the paths to all the available nodes.

The ~Structure Sheet

This sheet determines the content of the excel file : one line (under the header) of this sheet will be associated to a Data Sheet.

Sheet name	Predicate	Internal Id SFormat	Sheet Type
Default	n:network or n:device or n:childPoint<-n:device		

This sheet has two columns:

- **Sheet Name:** The name of the Data Sheet (the name of the tab it's associated to)
- **Predicate:** A NEQL query which will determine which components will be imported/exported in the associated Data Sheet.
- **Internal Id SFormat:** Optional value. This value allows the customization of the Internal Id of the rows in the associated Data Sheet during the export. If the cell is empty, the displayName of the component will be used as a base for the id.
- **Sheet Type:** optional value. Can be left empty or given the value "Components" to work with components. Can be given the value "Model" in order to work with aspects, nodes, node definitions, etc. See [Excel features](#) for a preview of the available feature for each sheet type



This sheet is mandatory to correctly import or export your components. Each line after the header will be associated to a tab in the excel file.

The Internal Id SFormat has been introduced in the 50.0.23.0, to see it, you must update your default config files.

If a component is exported in multiple columns, the internal id will be the one of the first sheet containing this component. To keep consistent ids in your sheets, use the same SFormat in each cell, or make sur that each component is exported only once

Example: If we have 3 lines in the ~Structure sheet: Network, Device and Point:

Sheet name	Predicate
Network	n:network
Device	n:device
Points	n:point and n:parentDevice->n:device

During the import, the system will create three tabs in the excel file.

Network	Device	Points
---------	--------	--------

If we want to add FolderPoint in our file, we only add a new line in the ~Structure Sheet.

IMPORT

Before importing the Data Sheets, the system reads the ~Structure sheet from the import sheet.

- If there are more Data Sheets than there are lines in the #Structure sheet, new lines will be added. In these lines, the sheet name will be the name of the tab and the associated predicate will be generated using the following format: xl:<Sheet name>. Example: Device Folder : xl:Device Folder. This predicate is a tag which will be added to each component of the sheet so they can be exported later.



To write the line, the import file must be closed.

- If there are more lines in Structure Sheet than tabs, those lines will be ignored.

! If the sheet is absent, the operation is cancelled

EXPORT

Before exporting the Data Sheets, the system reads the ~Structure sheet from the destination file. If the sheet is absent or empty, the system reads the sheet from the Config File (that you can select in this Export Tool Pane).

- If there are more tabs than lines in Structure Sheet, those tabs will be ignored
- If there are more lines in Structure Sheet than tabs, a Data Sheet will be created using a ~Default Sheet. The one from the destination file will be taken in priority, if absent, the one from the Config File will be used.

! If the Default Sheet is absent in both files, the operation is cancelled

The sheets after the export are ordered in this way:

- Sheets which have a line in the structure sheet (in the same order as the lines)
- Your helper sheet
- The standard helper sheets

The ~Variable Sheet

The variables are used to create shortcuts to save some time when you fill the Excel file.

1	General		Alarms		Histories		Conversion	
2	Artifact	Value	Artifact	Value	Artifact	Value	Artifact	Value
3	\$ModbusAsyncNetwork	modbusAsync:ModbusAsyncNetwork	\$OutOfRange	alarm:OutOfRangeAlarmExt	\$NumericInterval	history:NumericIntervalHistoryExt	\$Reverse	ReversePolarity

We use a notation with a \$ character as a prefix. A variable is a key:value entry that we define in the ~Variable Sheet. You can reuse the variables in the Data Sheets using only the key.

Example: `$ModbusAsyncNetwork = modbusAsync:ModbusAsyncNetwork`

There can be 4 different groups of columns

- **General:** Niagara TypeSpecs, Absolute slot paths or encoded facets
- **Alarms:** Alarm TypeSpecs
- **Histories:** History TypeSpecs
- **Conversion:** Conversion keywords

! You cannot put an id inside any variable value