

AddPxView

Summary

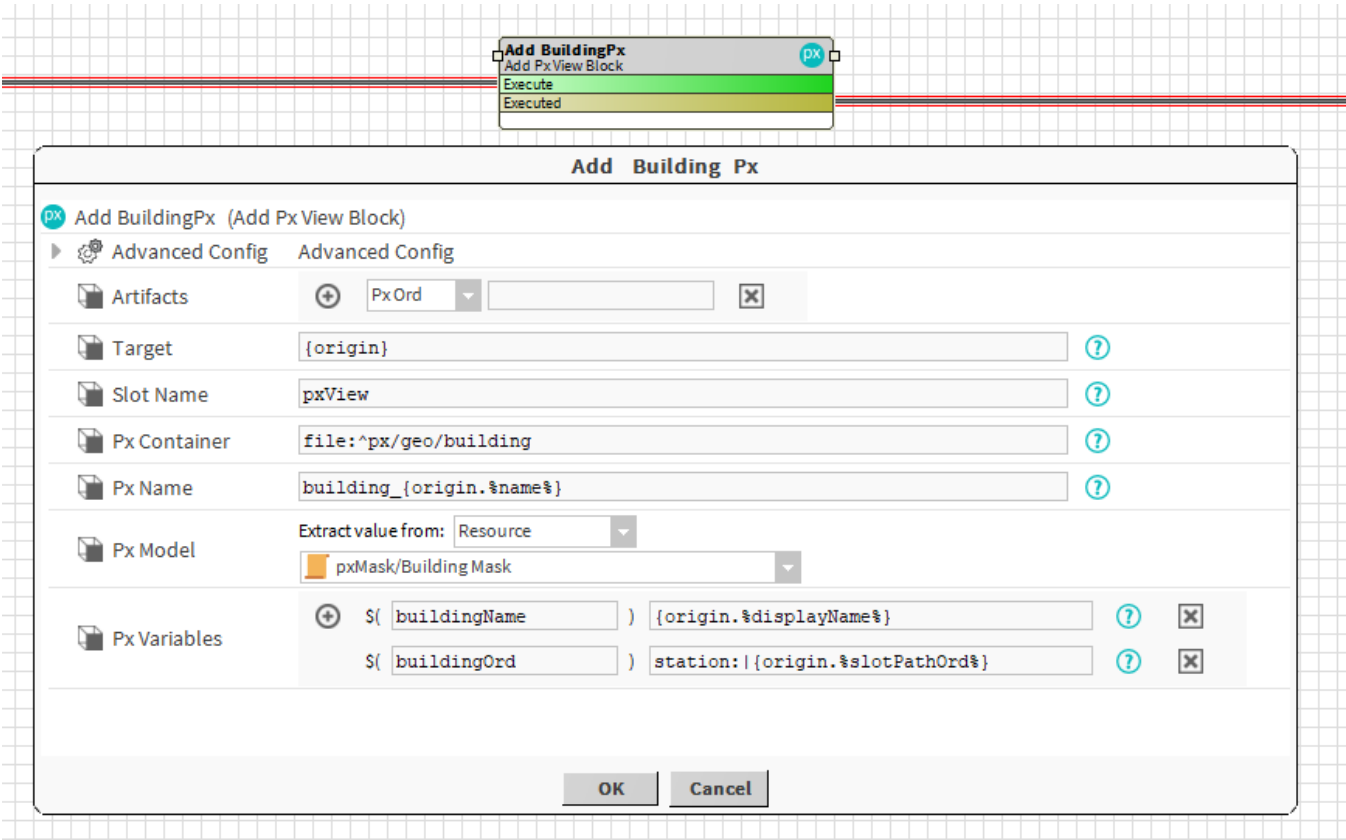
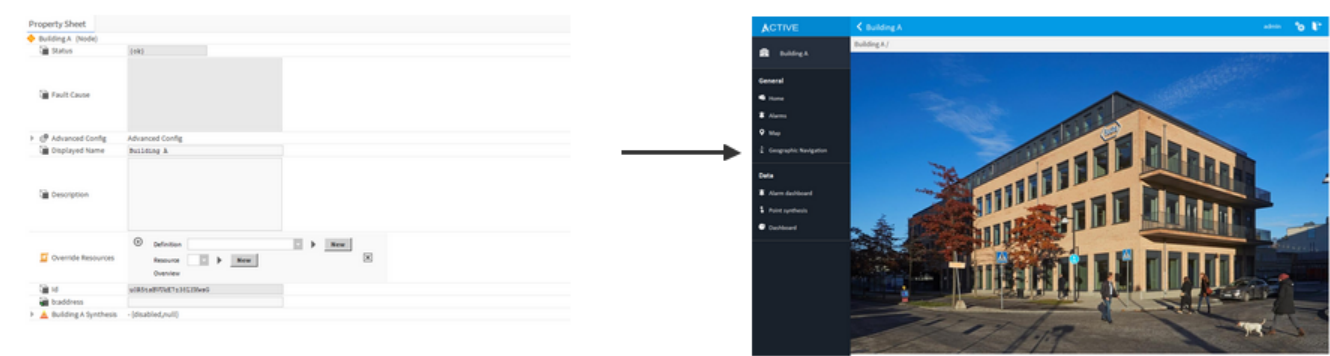
This block is used to create a PxView on a component. It creates the px and set it as a view on the component.

Implementation

- Drag and drop the block from the palette
- Link the "execute" action from the "executed" topic of the previous block

Example

This block creates a px (from a model) and sets it as a view on the component which represents the building.



Properties

- *Artifacts*: Artifacts created by this block

- *PxOrd*: ord of the created px . Ex: file: ^px/geo/buildingA.px
- *Target*: [SFormat](#) to define the component that will get the new view. Ex: the node building -> {origin}
- *SlotName*: [SFormat](#) to define the name of the pxView slot to add. Ex: pxView
- *PxContainer*: [SFormat](#) to define the folder (in the files) that will host the px file. If the folder doesn't exist, it will automatically be created. Ex: file: ^px/geo. Folders px and geo will be created if necessary.
- *PxName*: [SFormat](#) to define the name of the px. Do not add the extension ".px" at the end. Ex: "px Building A Cité" will be transformed into px_Building_A_Cite.px"
 - *FileName*: Names will be formatted as a file name (all special characters are replaced by _ and accents are removed)
- *PxModel*: [Resources](#) to define the model to use to create the px. See [General strategy parameters](#). Ex: all buildings will be created according to the same px model (which contains the name of the building at the top, left and top menus, a picture etc.)
- *PxVariables*: some variables can be passed to the px when the px is created and updated. Px variables can be set to every widget text field and ord field using the syntax \$(myVariable). Therefore it's more flexible than variables used for PxInclude (they are restricted to bindings only, here it will also work on the widget itself such as the text property). Ex: the model px for buildings will be personalized to add building's displayName and ord because we use \$(buildingName) and \$(buildingOrd) in the px

AdvancedConfig

- *SyncPolicy*: The policy for the creation of the px.
 - *Automatic*: the engine automatically updates only some parts of the existing px according to changes on the model, parameters and variables
 - *AlwaysReplace*: at each execution of the strategy, the px will be created from the model and it will override all previous changes
 - *NeverReplace*: the px will be created at the first execution of the strategy and will never be updated
- *IgnoredProperties*: list of slot name separated by semi column ; which should be ignored in the Automatic mode. Example: When you use this block to create a floor, there is a high probability you will change the background to put a floor plan. So on Automatic mode, you don't want your background to revert to the original, so you can add background to the ignoredProperties.

Behavior: DO

It creates the px,

Behavior: UNDO

It deletes the px and removes the slot from the component which had the view