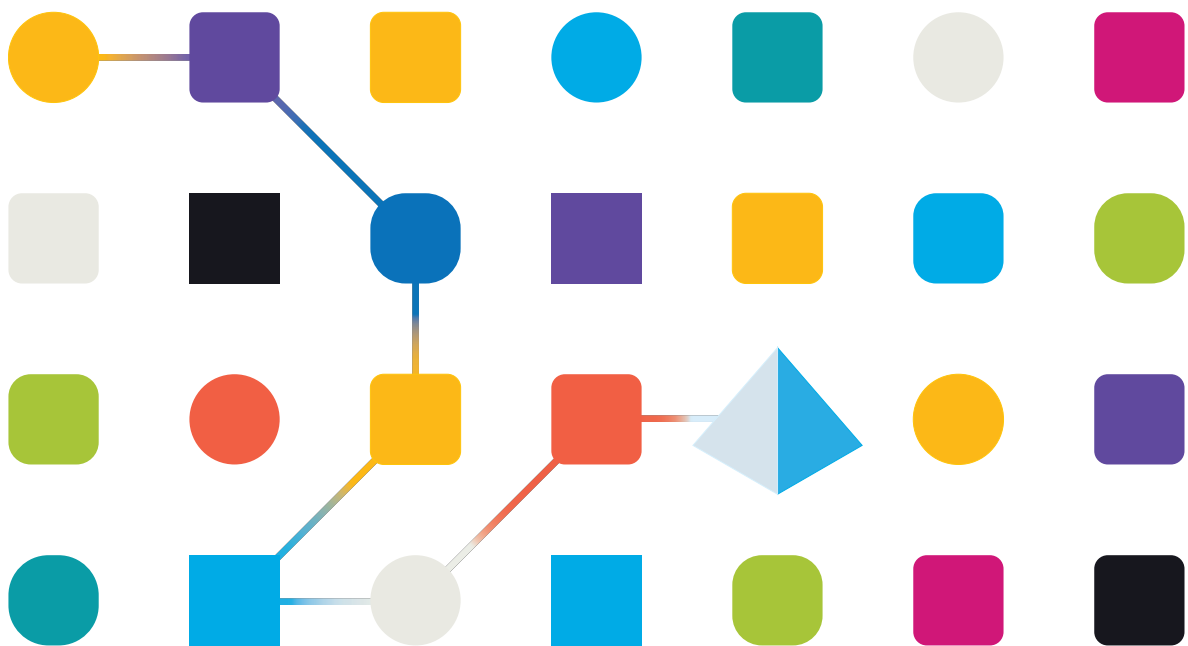




# Wireframer 5.0

## User Guide

Document Revision: 1.0



## Trademarks and Copyright

The information contained in this document is the proprietary and confidential information of Blue Prism Limited and should not be disclosed to a third-party without the written consent of an authorized Blue Prism representative. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying without the written permission of Blue Prism Limited.

### © Blue Prism Limited, 2001 – 2023

“Blue Prism”, the “Blue Prism” logo and Prism device are either trademarks or registered trademarks of Blue Prism Limited and its affiliates. All Rights Reserved.

All trademarks are hereby acknowledged and are used to the benefit of their respective owners. Blue Prism is not responsible for the content of external websites referenced by this document.

Blue Prism Limited, 2 Cinnamon Park, Crab Lane, Warrington, WA2 0XP, United Kingdom.  
Registered in England: Reg. No. 4260035. Tel: +44 370 879 3000. Web: [www.blueprism.com](http://www.blueprism.com)

# Contents


Wireframer .....	4
Getting Started .....	5
Creating a new wireframe .....	6
Saving .....	11
Editing a draft wireframe .....	11
Deploying a wireframe .....	11
Completing a wireframe .....	13
Spying .....	15
Attach page .....	16
Init page .....	16
Activate page .....	17
Wireframe action page .....	20

## Wireframer

Wireframer allows you to efficiently define business objects that can be used as part of an automation process. The benefits of designing using this methodology is that it allows the Automation Designer to rapidly deploy business objects and actions that will form the structure of the business process being automated. The simplicity in the plugin allows the definition of these business objects and actions, along with best practice techniques, to ensure that enterprise grade automations are always built.

This user guide covers Wireframer, a plugin within Hub which is part of the Blue Prism platform.

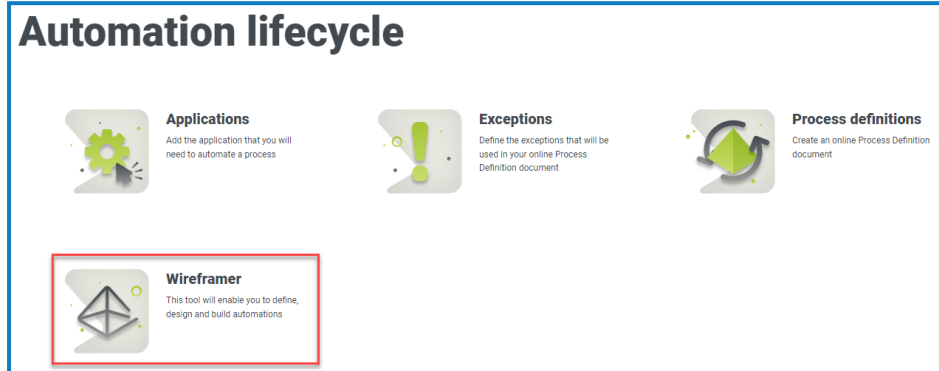
Wireframer can be found under the Automation Lifecycle Management (ALM) section of Hub. By utilizing Wireframer, automation developers can create a skeleton of the automation which will adhere to Blue Prism's best practice. It is assumed as part of this guide that the user is familiar with Blue Prism digital workforce and has experience with components such as Hub and Blue Prism.

 To use the full functionality of Wireframer, Blue Prism 6.4.0 or later is required.

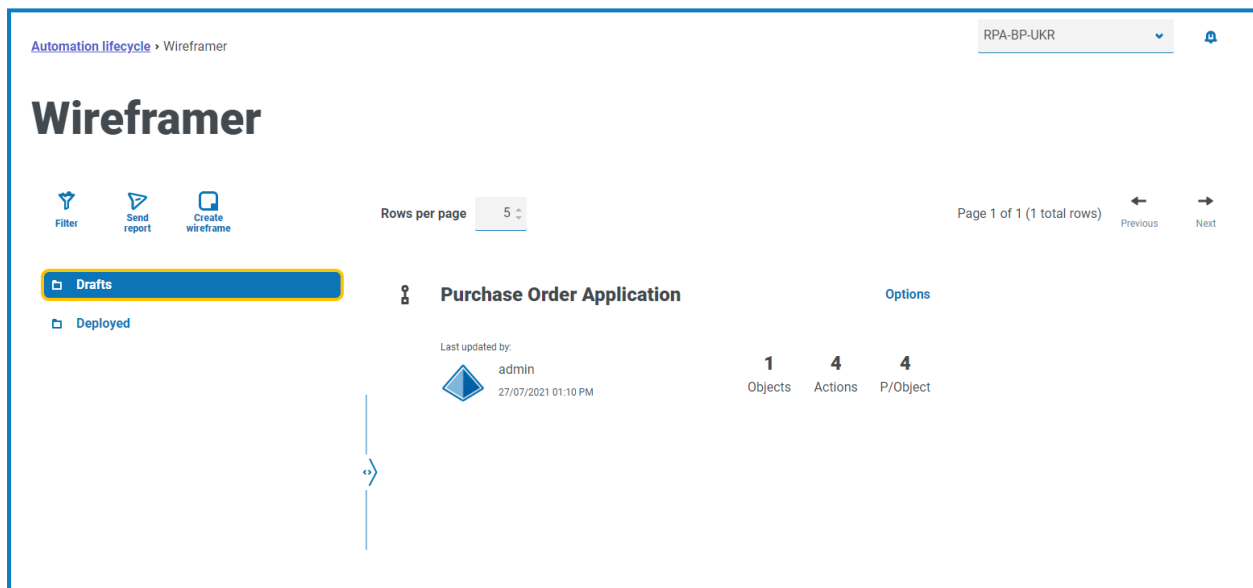
## Getting Started

Wireframer is a plugin that is installed into Hub. Like all plugins Wireframer will need to be installed by your Hub administrator before it can be made available to your user community.

After installation, click **Automation lifecycle** from the navigation menu and select the **Wireframer** card.



Once launched, the Wireframer home page displays. From here you can create a new wireframe, look back on previous deployed wireframes, or to continue where you left off with your draft wireframe designs.



## Creating a new wireframe

By selecting the **Create wireframe** option on the Wireframer landing page, you are prompted to enter a name and a description for the new wireframe.

Enter the:

- **Wireframe name** – The wireframe name is used to enable easy identification of the wireframe using the search and filtering capabilities of Hub. This enables the organization to identify wireframes when saved as draft or deployed.
- **Wireframe description** – This is optional, however, it is best practice that this is supplied to ensure that if others are working on the same project they understand what the wireframe you have created relates to. The description enables a user to view the details of the wireframe without needing to open the wireframe and view each individual object, so a good description is really essential.

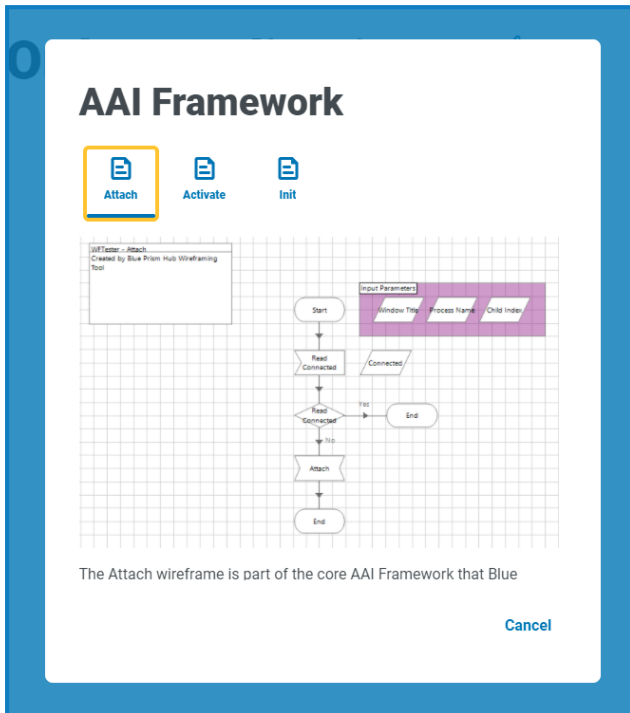
Once the form has been completed, click **Create wireframe** to launch Wireframer.

The created wireframe enables you to define objects and actions, the same principle and manual techniques that are used in every automation build. At the top level, objects can be added.

After you have added objects you can then add actions to those objects, defining the capabilities these objects can offer to an all-consuming automation component or process. Across the top of the Wireframer, a tally is kept of the objects and actions that the wireframe consists of, in addition to the average actions included within each object.

To update the wireframe name and description, click **Edit**, update the details and click **Save**.

Three actions are automatically added for any new object. You can view the actions by clicking **View AAI** when creating the wireframe.



All objects that are defined within the Wireframer follow Blue Prism best practices. In summary, this creates three actions by default:

- **Attach** – This action creates the attachment relationship between the object and the application.
- **Activate** – This action will bring the application window to the foreground ensuring that global mouse clicks or keyboard events are directed at the focused application.
- **Init** – This action ties together the Attach and Activate Actions so they can be called at the top level of any subsequent Action.

To build your wireframe process, click **Create object**. This displays the Create a new object form.

Enter a name and a description of the object and select whether the Application Model Type is a Windows or Browser based application that you want to interact with. This is used to construct a basic Application Model within the object which can then be used.

## Create object

✕

Close drawer

**Object name \***

**Object description**

**Application model type \***

Windows

Browser

Save

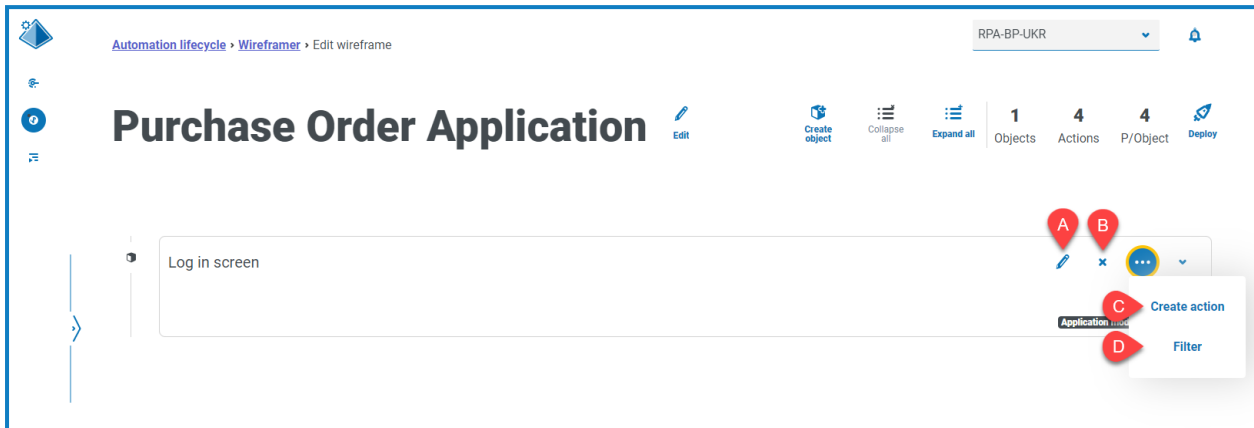
Note: All objects generated will come with **Attach, Activate and Init** actions.  
Click the button below for more information:

View AAI

If you consider the design of an automation process, you would normally start with 'launching' the application. When using Wireframer, this step is built separately to the wireframe. The use of Wireframer is to map the screens within the application that you want to interact with using the different structural types of design.



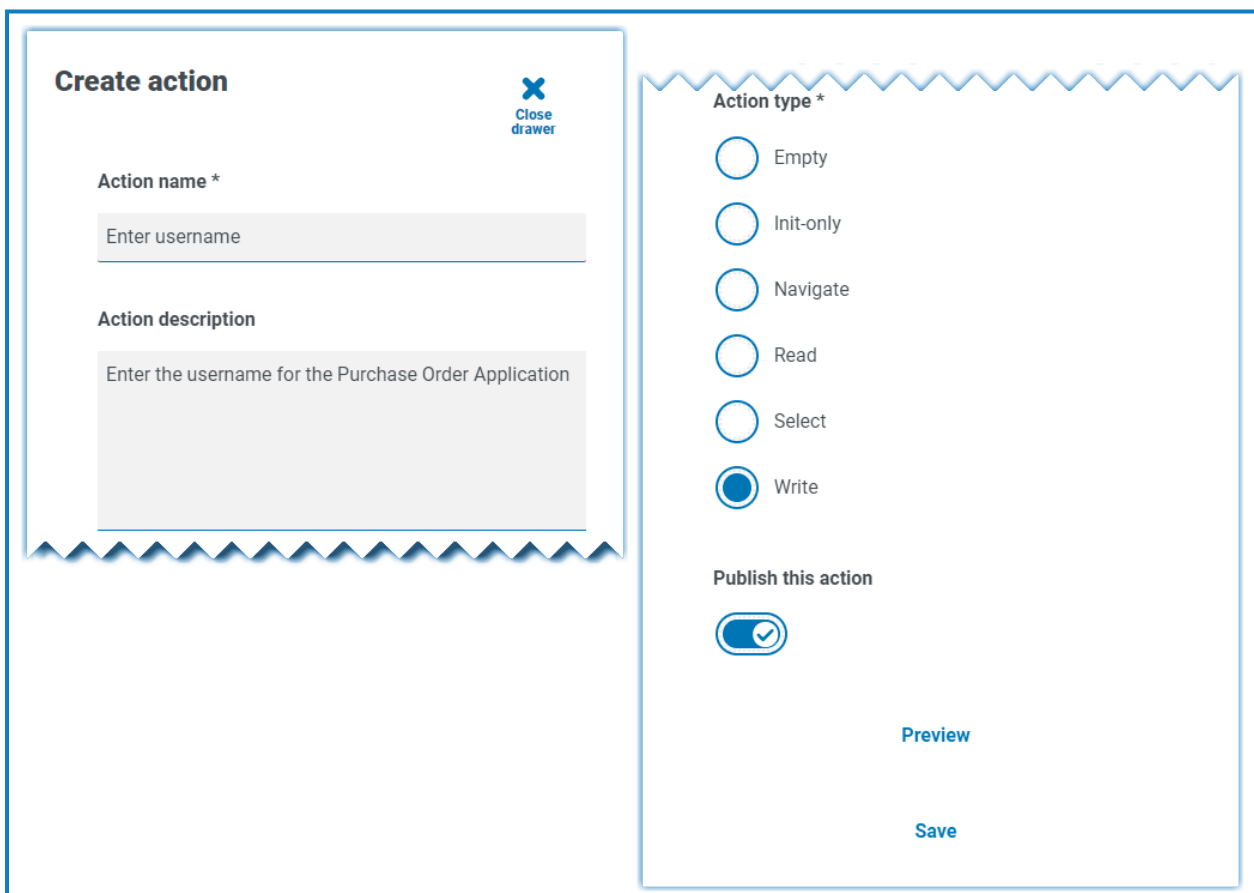
Click **Save** to create the object. The Wireframer is updated with the information provided and the tally is updated to reflect the new addition.



The following options are available for a created wireframe:

- A. **Edit** – Update the object name, description, and application model type.
- B. **Delete** – Remove the object from the wireframe.
- C. **Create action** – Add a new action. Click the ellipsis icon to see this option.
- D. **Filter** – Filter by one of the available attributes to help you locate the required action. Click the ellipsis icon to see this option.

Select the Create action option to create an additional action alongside the existing default actions. Like the earlier forms, enter a name for the action and a suitable description, then select the type of action it will perform.



There are six available action types:

- **Empty** – An empty action that has no inbuilt functionality added to it, it is simply a Start stage linked to an End stage.
- **Init-only** – An Init action which is a Start and End stage with a Page Reference stage in between that references the Init action. This will ensure that the application is attached and the window is focused on the screen.
- **Navigate** – Deployment of a Navigate stage so you can associate it with a button push or click.
- **Read** – A Read stage that enables the reading of information from the application.
- **Select** – Is a Navigate stage with an Input, that enables you to access drop-downs or check boxes where you can pass a true or false flag to 'tick' the required box for example.
- **Write** – A Write stage enables the population of a field in an application form.

Finally, after the form has been configured for the action you require, select **Publish this action**. If this is not selected when deploying, the created actions in the object will not be available to the object when you come to use it in your processes, so make sure that the actions are selected to publish before deploying.



The AAI actions are not published. This is because you only need to call them within the object itself, so it is a local interaction, therefore there is no need for them to be published.

- The Init action calls the Attach and Activate actions (internally within the object).
- Your actions calls the Init action (internally within the object).
- Your actions are published meaning they are exposed for other processes or objects to use those actions.

To view the action you have added to the object, click **Preview**. This displays the selected action type, for example, the Write action.

The Write wireframe uses a Write stage for the 'Interaction' stage and allows for an input data item to be used as an input to the Write stage. This wireframe type should be used for thick client applications where write stages are expected and web application form elements.

To close the preview, click **Cancel**. Once you have configured the action, click **Save** to store the changes. As with the previous option menus, the option button allows you to Edit and Delete the action as required.

You can then repeat the steps for adding other actions as required to your object, or add additional objects and actions as required to map out a particular application or section of an application as required.

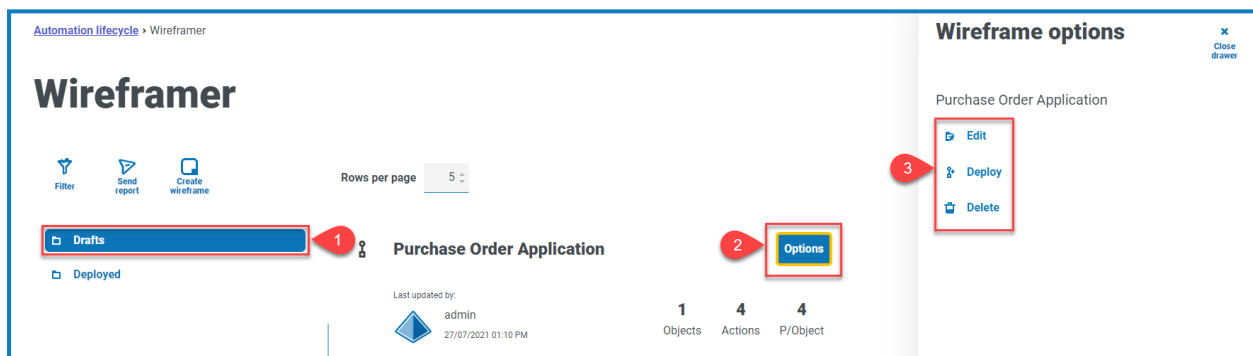
## Saving

Any changes you make to the wireframe as you build out the objects and actions are automatically saved. If you close the window, the wireframe is saved as a draft and you can resume from where you left off.

## Editing a draft wireframe

Any wireframe that has not yet been deployed is stored in the Drafts section of Wireframer. To edit a draft:

1. From the main Wireframer page, click **Drafts**.
2. Click **Options** for the wireframe you want to edit.
3. Select the required option:
  - **Edit** – Opens the wireframe to let you add, edit, and delete objects and actions.
  - **Deploy** – Deploy the wireframe to the selected environment. The wireframe is moved from the Drafts area of Wireframer to the Deployed section.
  - **Delete** – Delete the wireframe.




## Deploying a wireframe

When you select **Deploy**, the following form displays:

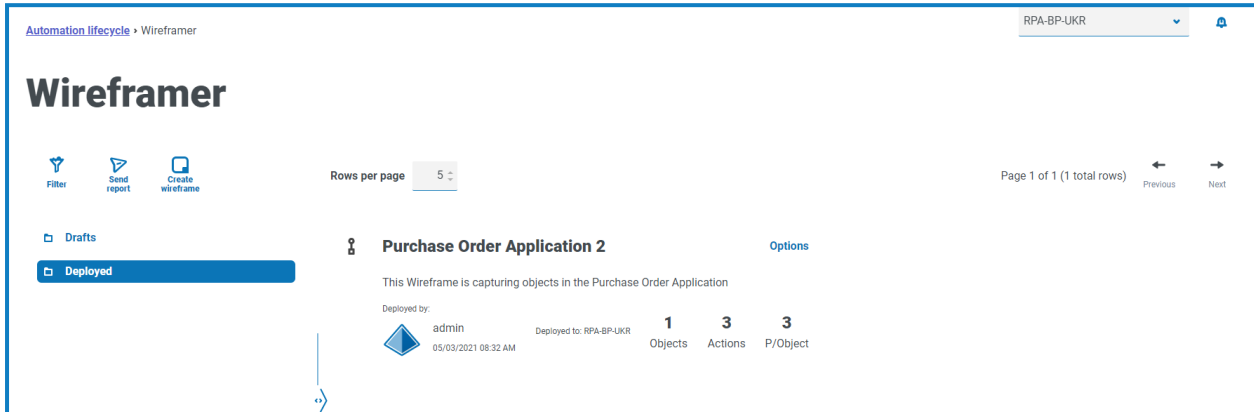
The screenshot shows the 'Deploy' form. At the top, it says 'Once deployed wireframe can not be modified.' Below this is a dropdown menu for 'Environment \*' with 'RPA-BP-UKR' selected. Underneath, it states 'This will create 1 object(s)'. At the bottom, there are two buttons: 'Cancel' and 'Deploy'.

This displays a summary of what will be deployed and allows you to select which environment you wish to deploy the object(s) into.

 You can also choose to Deploy from the Create Wireframe display if you have completed all the object(s) and action(s) in one session without needing to use the draft functionality.

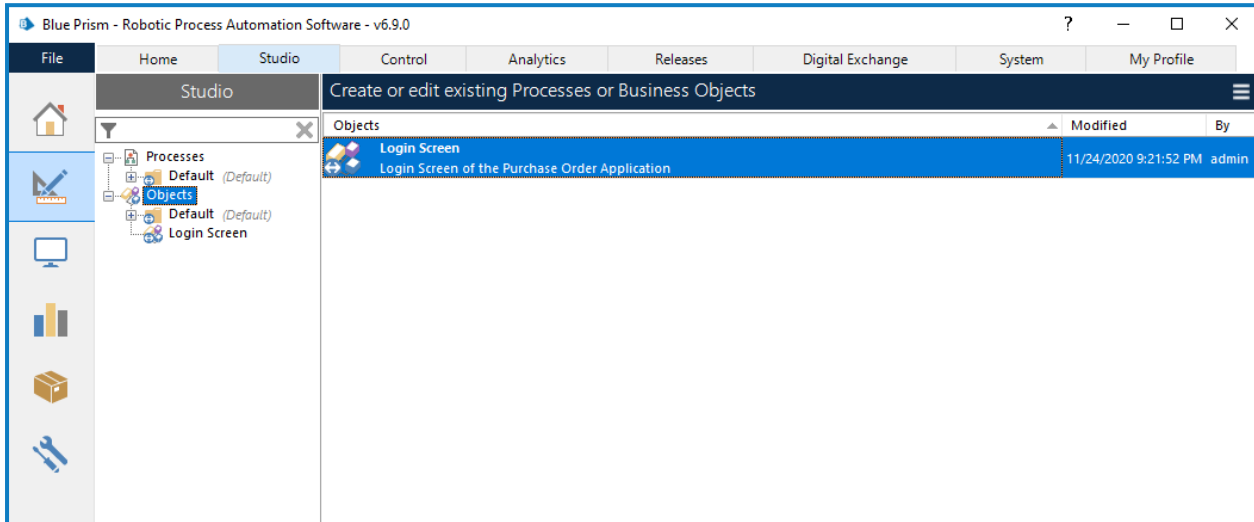
Clicking **Deploy**, creates the wireframe and deploys the object(s) to that environment. Once an object is deployed through the Wireframer it cannot be edited or deleted within Hub. The deployment is added to Blue Prism and is only available there for further additional work or, if required, deletion.

Once deployed, the object can be viewed in the wireframe deployment screen, accessible by selecting **Deployed** on the main wireframe menu.




## Completing a wireframe

The wireframe is a skeleton of an object and does not contain all the necessary information to perform the automation. If you switch to Blue Prism, you can see the deployed wireframe object. In the example below, the Login screen has been added.

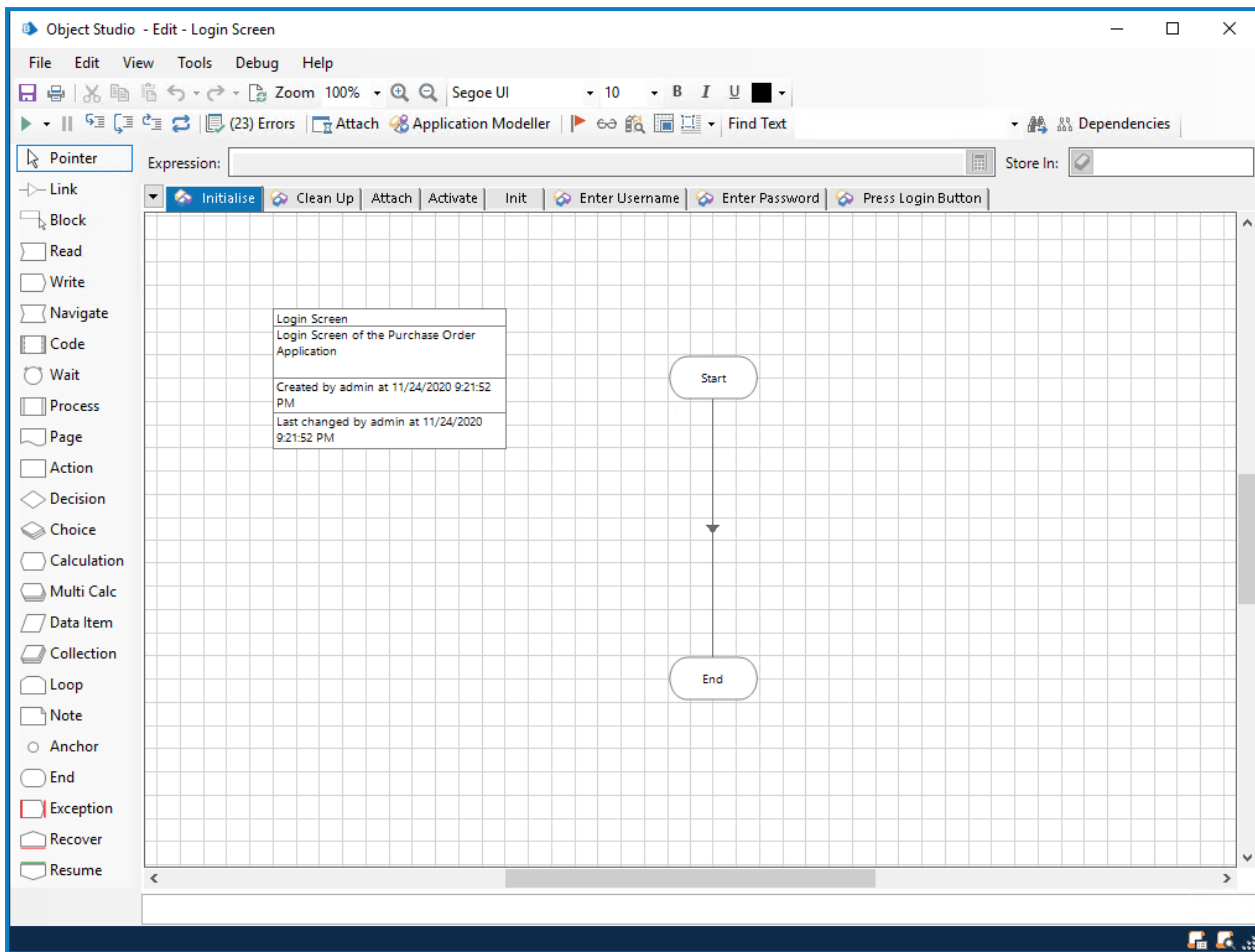


You should drag and drop the wireframe object into the respective Group folder before commencing any work.

 Only a Blue Prism System Administrator role can access objects at the top level of the hierarchy.

Objects need to be within Group folder before they can be worked on by other Blue Prism roles. Once within a Group folder, the Object Editor is launched by double-clicking on the object.

The image shows the object as designed within the Wireframer plugin.



The standard default pages for an object, Initialise and Clean Up are provided as standard, as well as the three default pages that were configured in Wireframer, which are the Attach, Activate and Init. You will also see pages for any actions you have added to the model.

## Spying

Before you can complete the pages and assign the properties, you need to 'spy' the application using application modeller. This way, all the elements can be captured that are needed to feed into the action stages as properties. This user guide does not cover 'spying' an application (information is available in the [Blue Prism Help](#)), however there are some key points when 'spying' that need to be performed.

On the majority of the actions that are created from Wireframer there are two action stages, referred to as primary and backup elements. When designing and 'spying' elements in the application modeller, primary and backup elements can be used. A primary element is usually an element in the application modeller which:

- Has been spied in the most accurate mode (for example, Win32 for thick client applications);
- Includes match attributes that specifically dictate where the application modeller can find this element in the application.

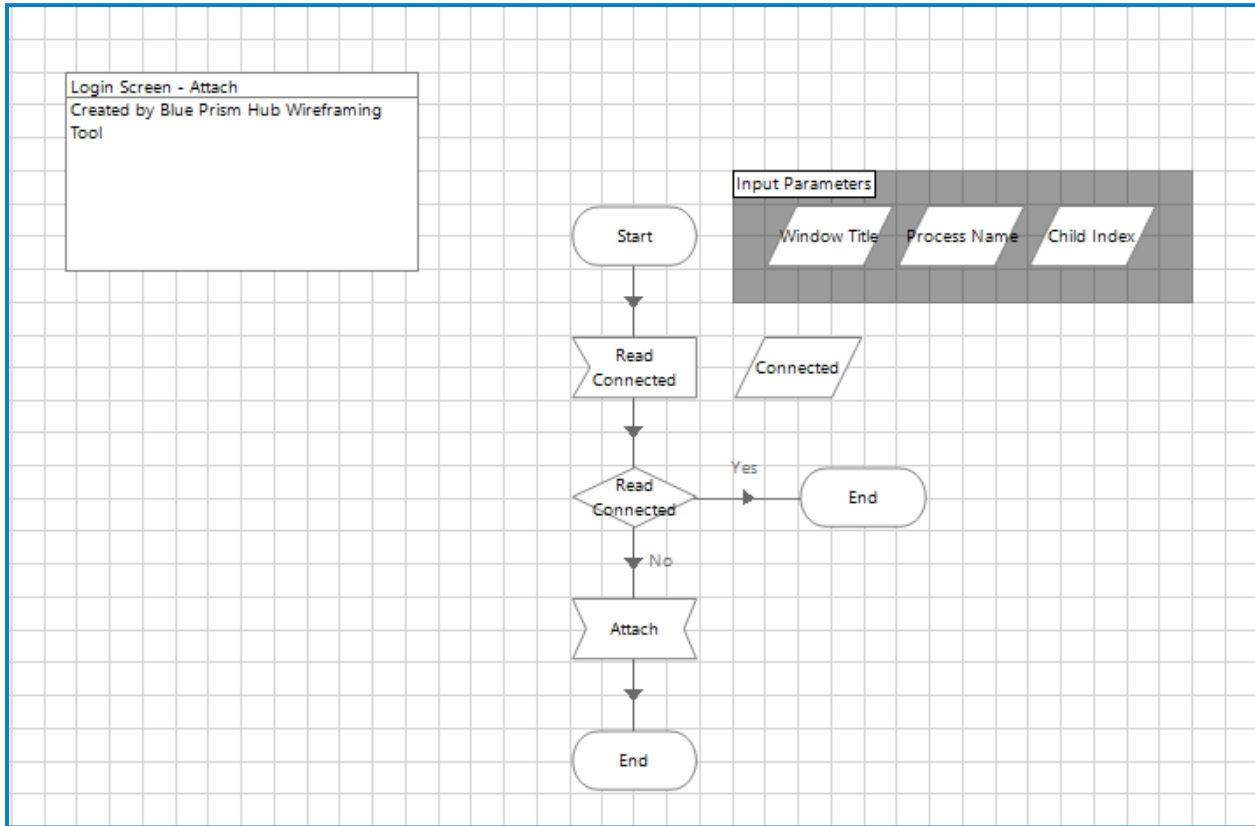
A backup element is usually an element in the application modeller which:

- Has been spied in the most accurate mode using an alternative method (for example, AA mode for thick client applications);
- Includes match attributes that don't specifically dictate where the application modeller can find the element in the application. The match attribute criteria require that the application modeller, search for the element within the DOM based on the match attribute selection, or search within the application element hierarchy API.

The primary and backup elements help to improve speed and resiliency of the object. Consider a case where an 'OK' button needs to be pressed on a web application. You would spy the web application using 'HTML' mode and use the 'Path' variable to match the attribute. However, if the web application is updated or due to additional HTML elements being added it has resulted in the spied 'Path' value becoming invalid, the web application button wouldn't be found. Therefore, a second spied version of the element would provide backup resiliency if this occurred. Once there are multiple versions of an element spied, you can leverage a wait stage that waits for either the primary or the backup element to exist. Whichever of the elements is found first, then that element is used to send a 'click' to (or appropriate action). Once all the elements have been captured, each of the pages can be updated using the 'spied' elements to complete the object.

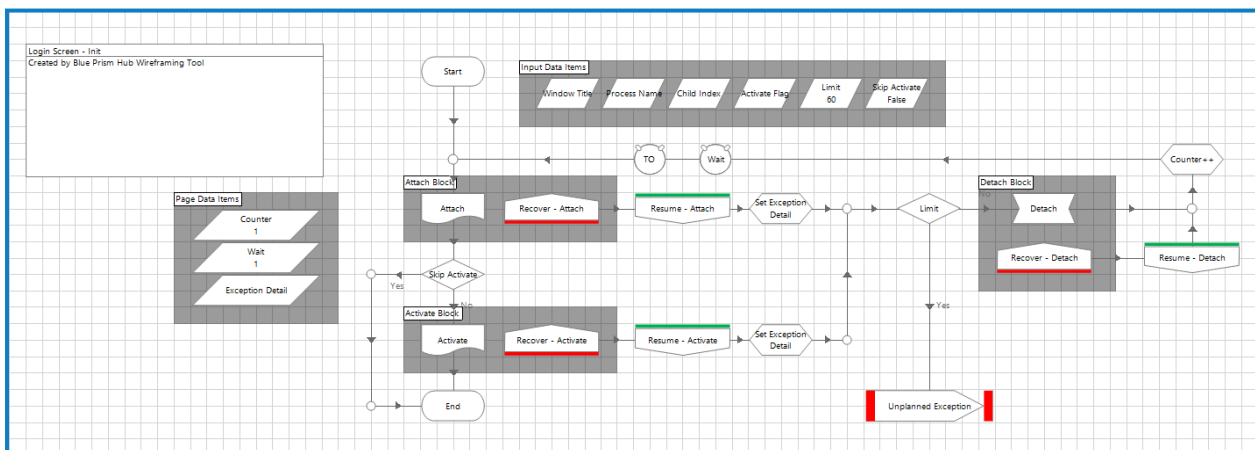
## Attach page

The Attach page is complete, and no further changes are required.



## Init page

The Init page is also complete, and no further changes are required.

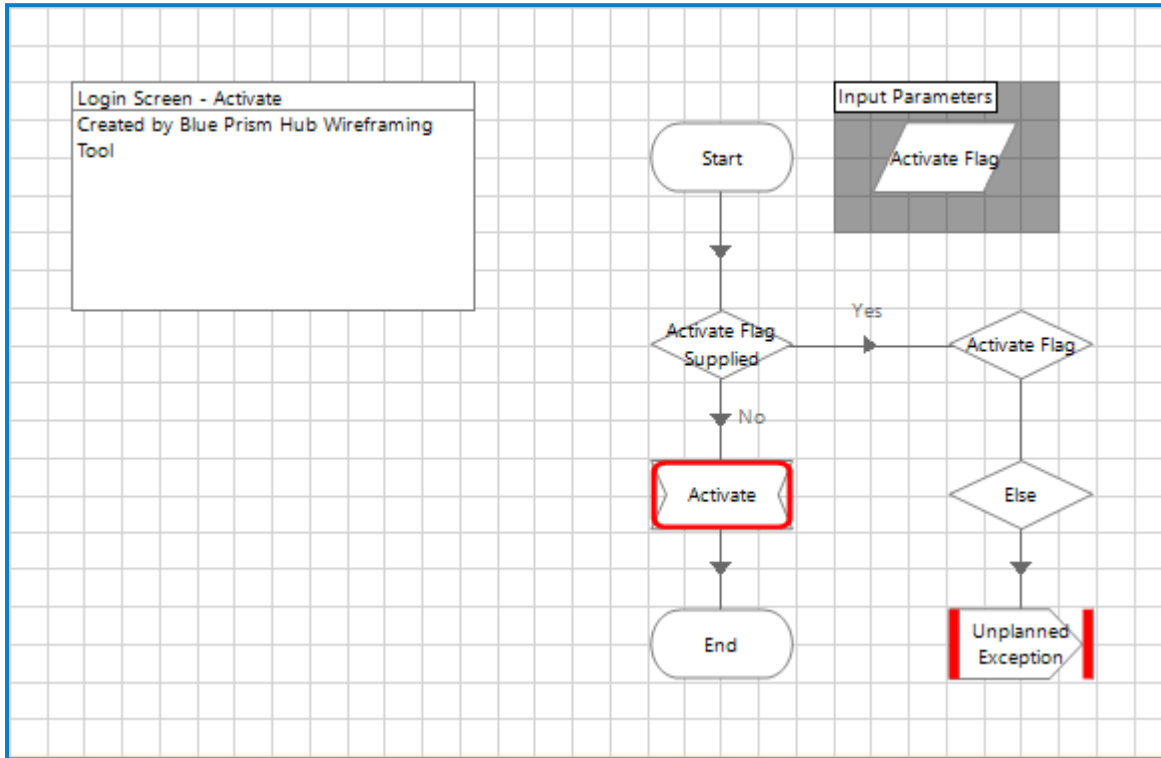


In the Init page, in line with best practice techniques, you can see the exception logic for recovery and resumption. This page, also as mentioned earlier, references the Attach and Activate actions through a Page Reference stage, to ensure that the application is attached, and the appropriate window is brought to the foreground and focused.



## Activate page

The Activate page does need an application element to be provided before it can be used, this is indicated by the red square, breakpoint, around the stage.



By double-clicking on the Activate Navigate stage you will see that details relating to the application element need to be provided to the stage.

**Navigate Properties**

Name:

Description:

**Application Explorer**

Filter the tree...

- Purchase Order Application
  - Main Screen
    - Username
      - UsernameBU
    - Password
      - PasswordBU
    - Login Button
      - Login ButtonBU

**Actions**

Element	Params	Action	Inputs Set
Main Screen	...	Activate Application	N/A

Pause After Each Step (timespan/secs)

Move Up Move Down Add Remove

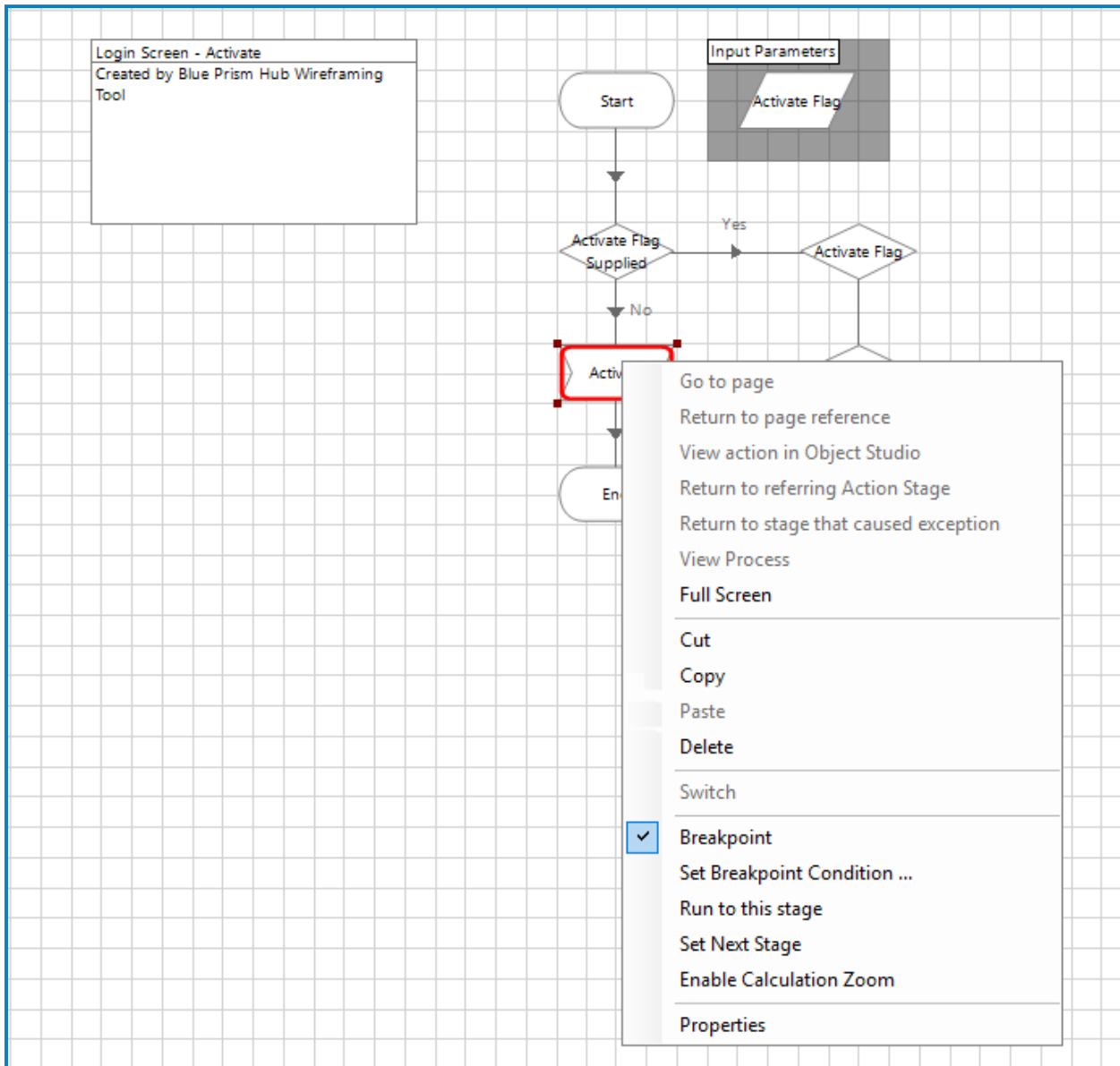
**Inputs**

Name	Datatype	Value

Stage logging:

Warning threshold:  Number of minutes  (0 to disable)

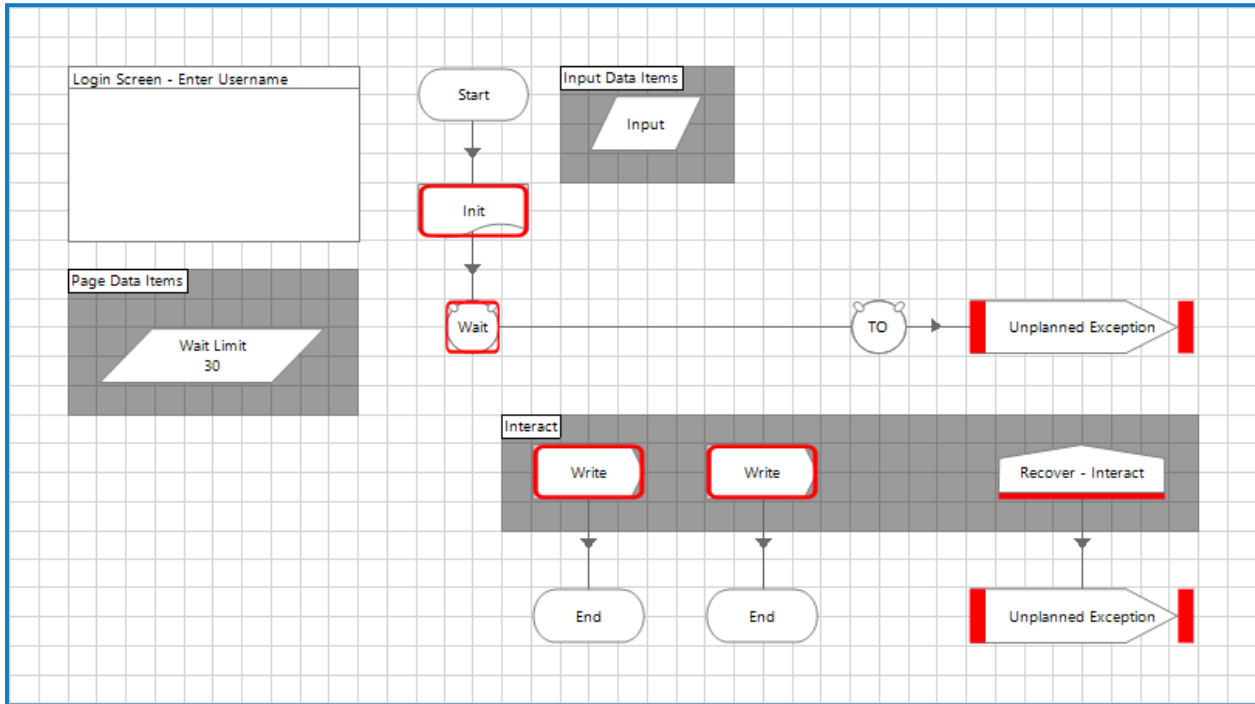
Once you have configured the properties to Activate the application window, the next step is to clear the Breakpoint; right-click the stage, and click **Breakpoint** on the shortcut menu.



## Wireframe action page

Using the Wireframer, the action stages that need to be configured can be reviewed. The information below uses a Write action as an example.

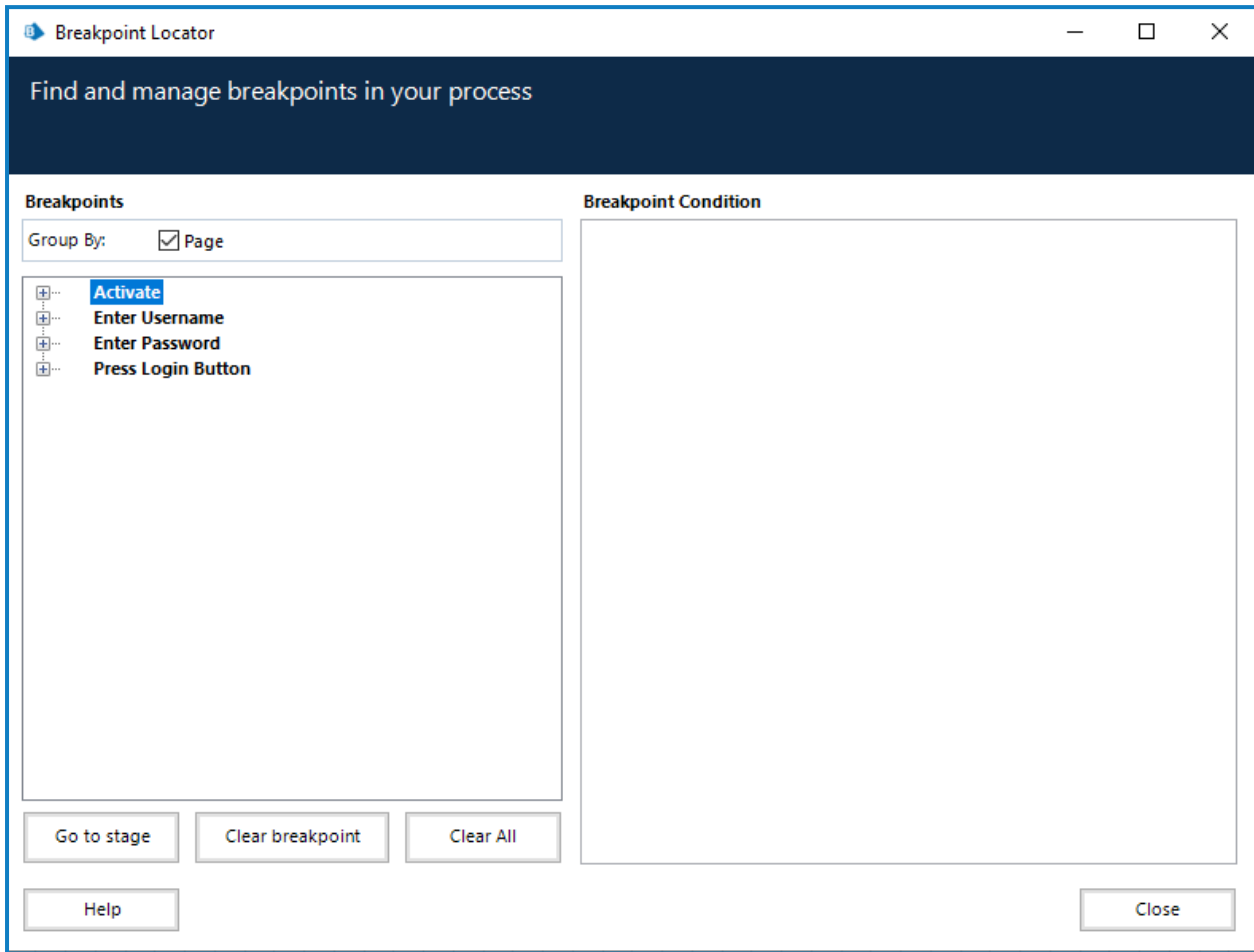
The Write Username page in this example has numerous stages identified that require properties to be set or application model elements to be aligned.



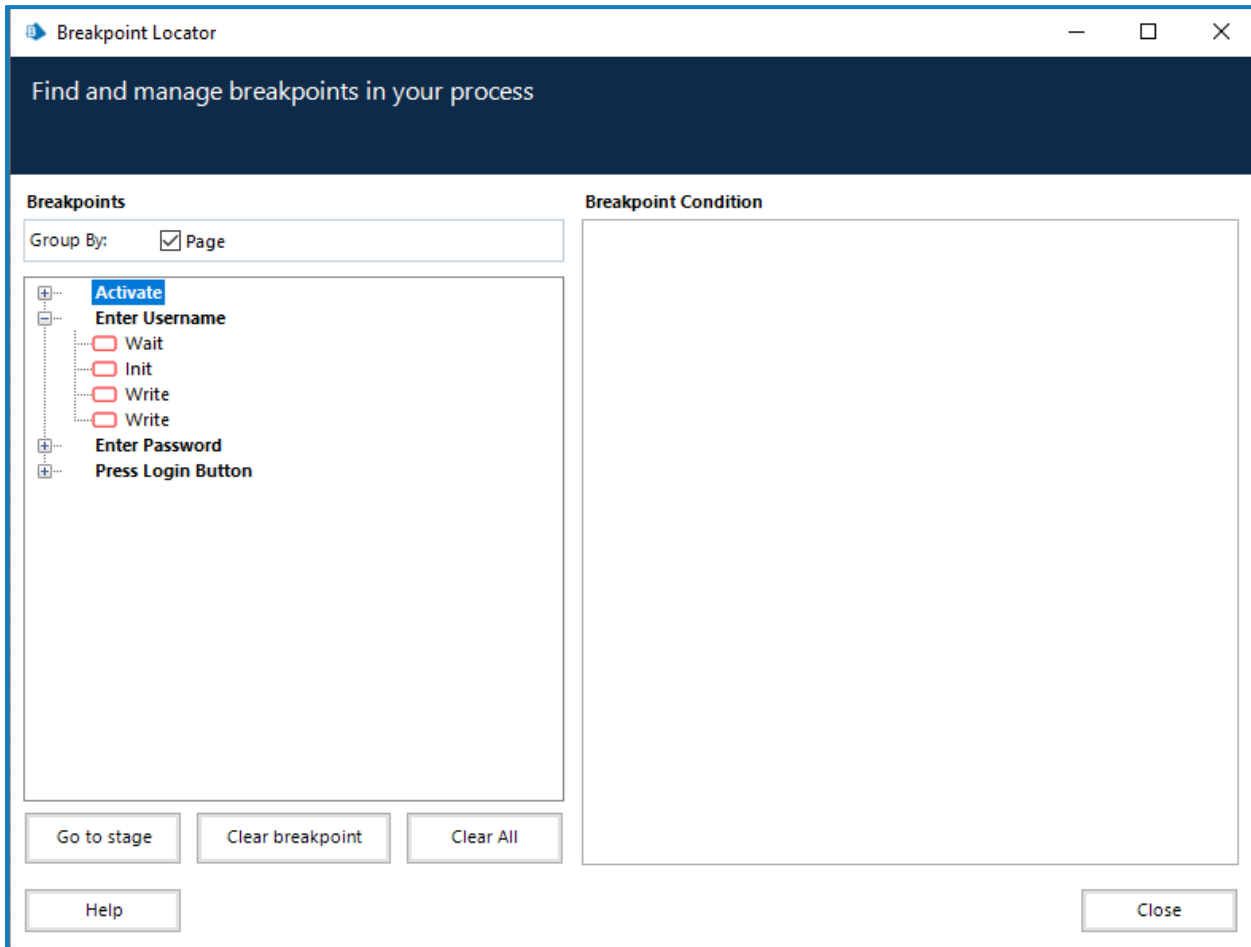
To work through the different stages that require attention, the View Breakpoints option can be used. This is activated by selecting the flag icon in the main menu bar.



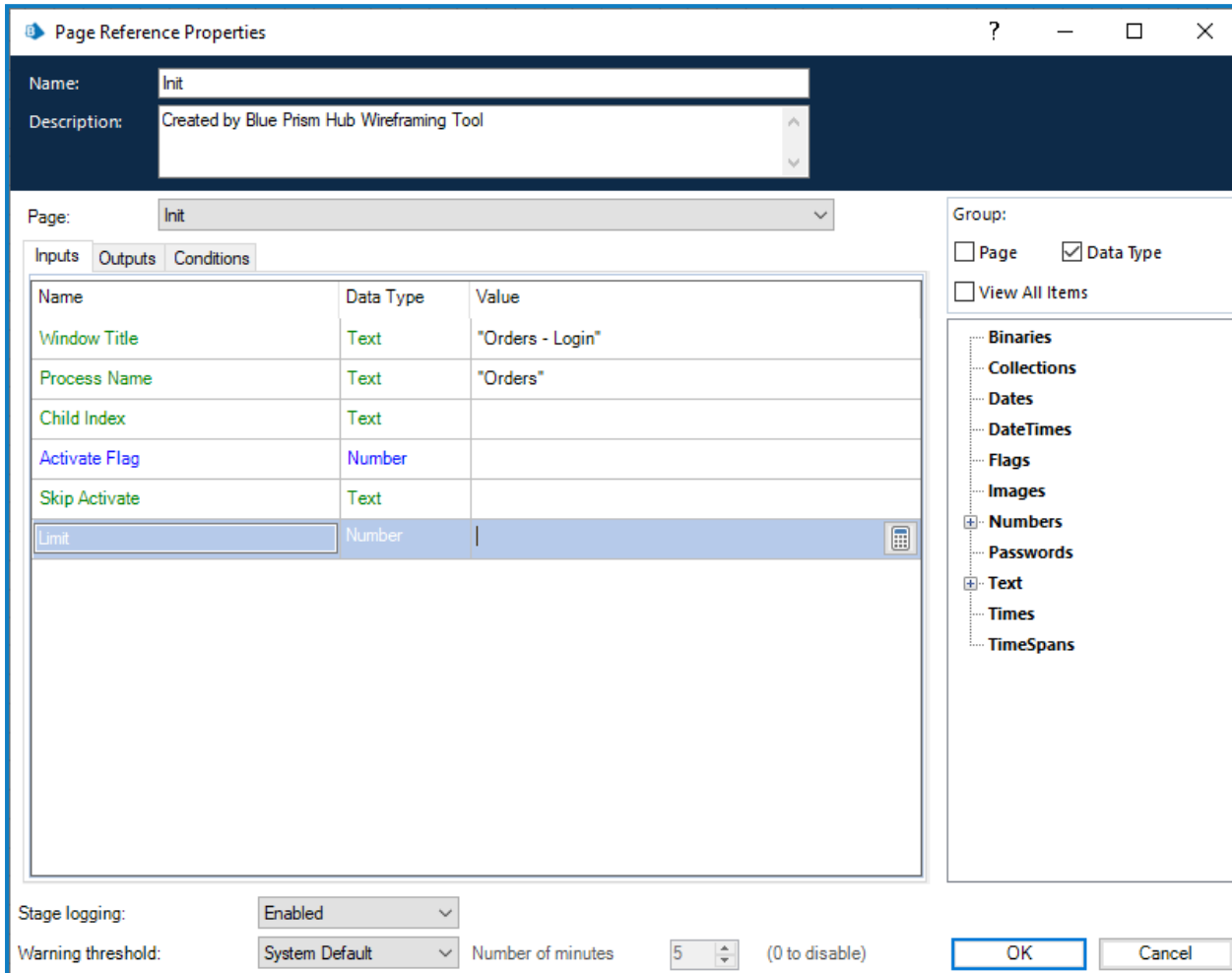
By selecting this command, the Breakpoint Locator screen displays which lists all the breakpoints set within the object.



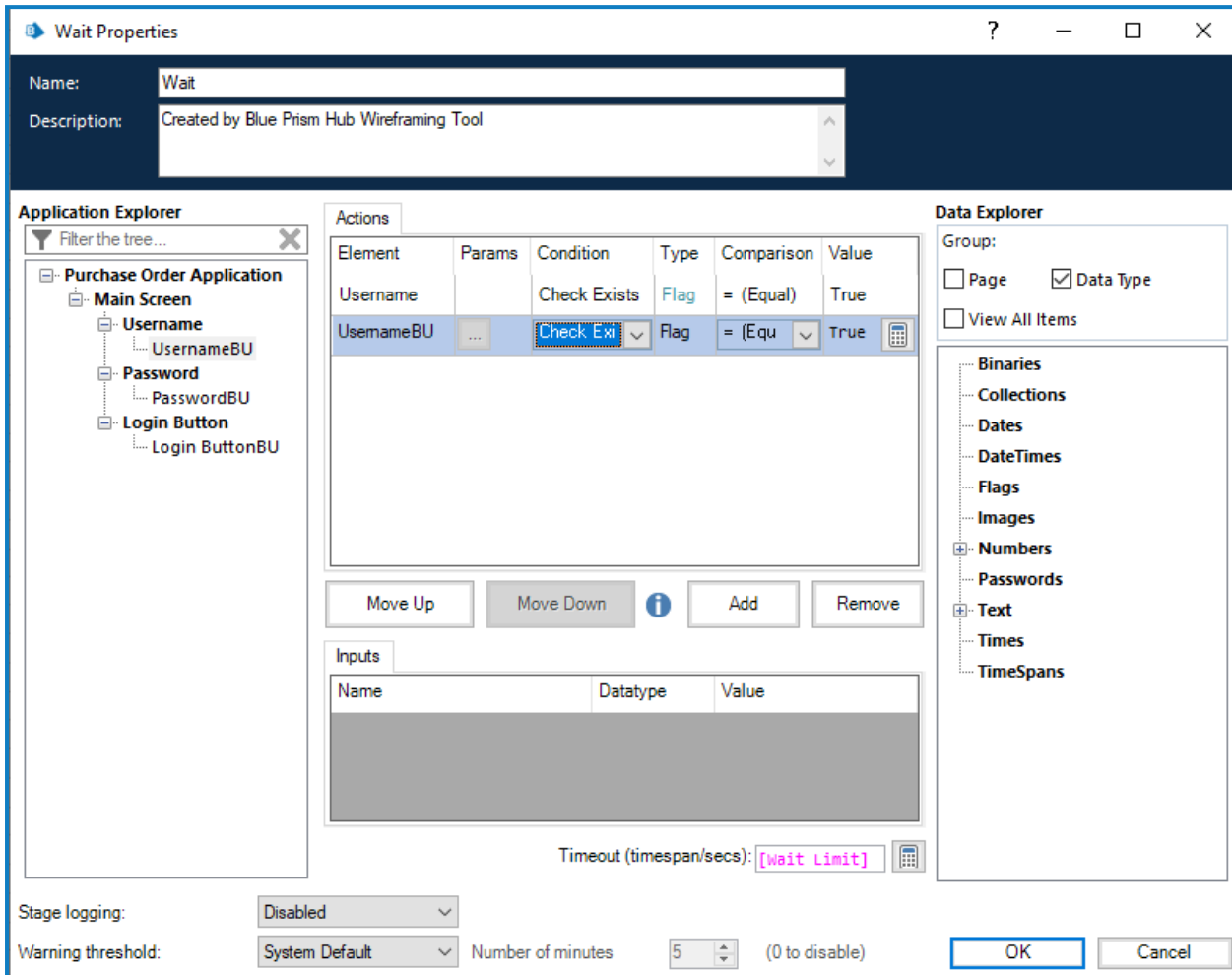
The screen lists all the breakpoints across the object pages, these can be expanded and then used to navigate to the appropriate stage on the appropriate page, by clicking **Go to stage**.



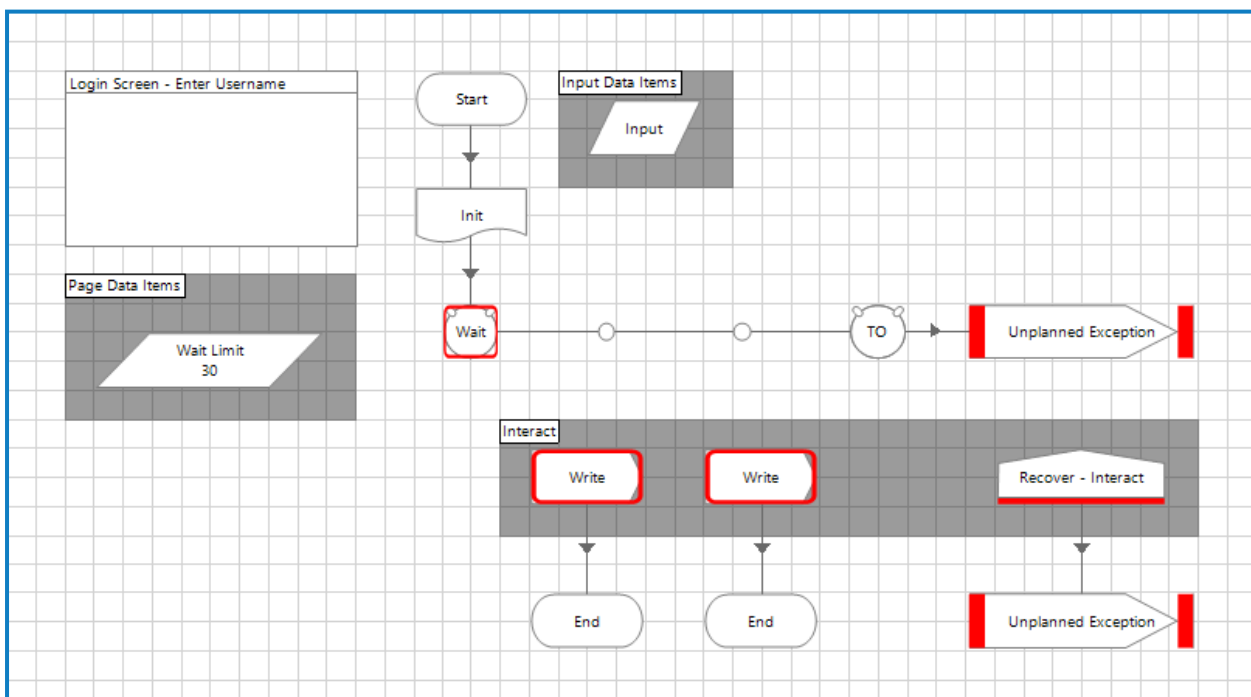
This screen also enables you to track your progress when adding properties and application model elements to all the stages that have breakpoints aligned to them. Once you have actioned all the changes required, the View Breakpoints command produces a clear screen with no breakpoints remaining. The Init stage should be configured to the specific application being used. Once applied, the Breakpoint is then removed.



When configuring the properties on the Wait stage it is important to add two element rows, so that you build the resiliency within the object. In the image below, two rows are shown configured with the two 'spied' elements.

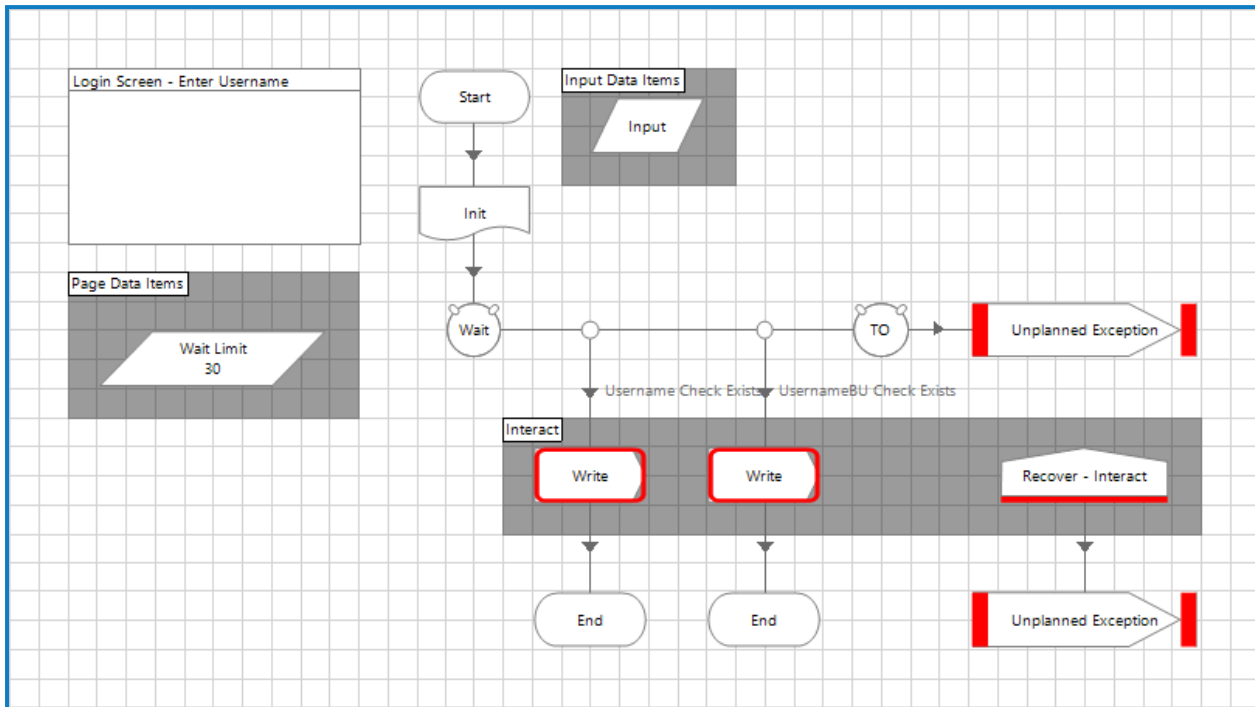


Once the properties are saved, two anchor points are added to the diagram which need to be connected to the two stages, in this example, the Write stages.





Once connected and tidied up, the Breakpoint for the Wait stage can be removed.



It is important that when configuring the Write stages that the correct elements are used which correspond to the two Wait stage conditions that were set, so in the example the left condition is 'Username Check Exists' so using the primary element. Whilst the right shows 'UsernameBU Check Exists' the backup element. The left Write stage is therefore configured.

Value	Element	Params	Type
[Input]	Username	...	Text

Name	Datatype	Value

Completing the second Write stage in a similar way but using the backup element, then removing the two Breakpoints completes this wireframe action page.

