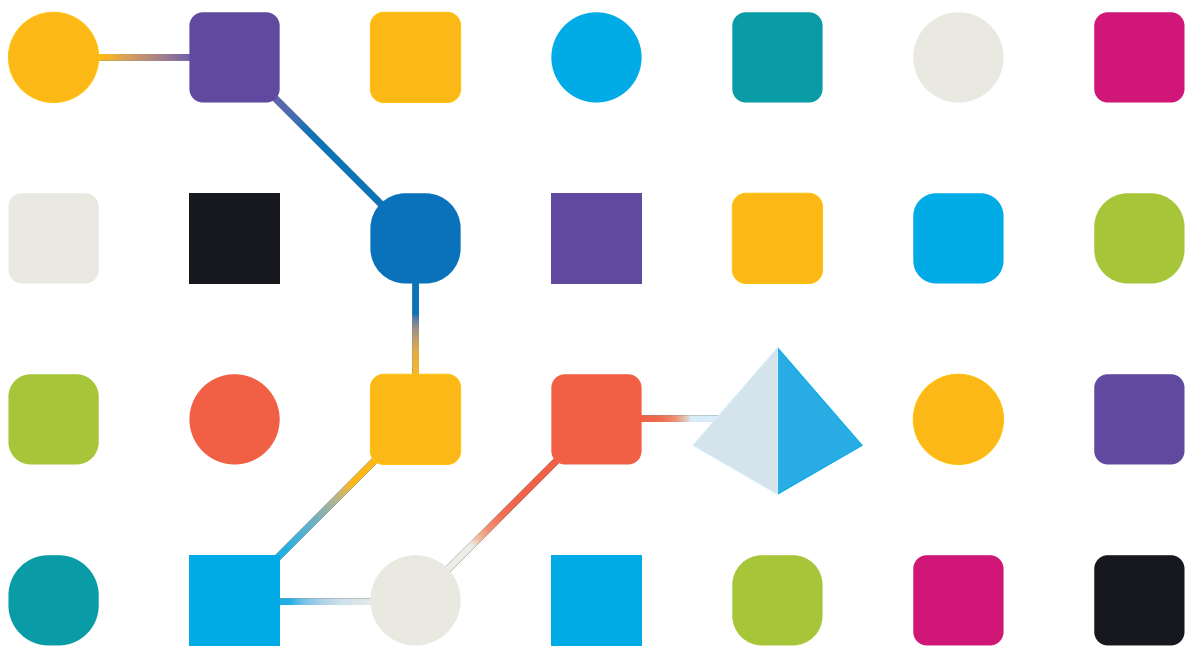




Blue Prism Desktop 1.1

Process Developer 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

Contents

Process development for Desktop	4
Desktopdebug mode	4
Process design elements for Desktop	4
Differences in Blue Prism functionality	14
Desktop process design considerations	16
Solution design considerations	16
Process structure	16

Process development for Desktop

Automations for use in SS&C | Blue Prism® Desktop can be developed using current SS&C | Blue Prism® Enterprise functionality and methodology. This guide describes the differences between a standard Blue Prism interactive client and a Blue Prism interactive client with a Desktop license applied, and outlines special considerations when developing processes for Desktop.

Before creating a Blue Prism process to run in Desktop, it is recommended that you refer to [Desktop process design considerations on page 16](#) for a suggested design approach.

Desktopdebug mode

If you are working in a dual license environment (one where both a Desktop and an Enterprise license are applied), you can design and test processes either for Desktop or Blue Prism Enterprise. When working with Desktop processes, click the **BPD** button in the Validation toolbar to switch on the Blue Prism Desktop debug mode.

When Desktop debugging is enabled, Process Studio works as if the environment has only a Desktop license applied:

- All Desktop-only functions (for example, the [Input stage on page 6](#), [Output stage on page 8](#), and [Enterprise Session stage on page 9](#) types) are available and are considered during debugging.
- In the Action stage, objects that are not relevant to Desktop are hidden.
- Warning messages in the [process validation tool](#) will notify you when any Blue Prism Enterprise functionality that is incompatible with Desktop automations is included in the process.

When Desktop debugging is disabled, Process Studio works as if the environment has only an Enterprise license applied:

- The Desktop-only functions are not available, and if the process was imported from a Desktop-only environment, error messages warn if the process includes any of these functions.
- All Action stage objects are available.

Process design elements for Desktop

Blue Prism process developers can create automations to run on Desktop. The processes can incorporate forms (using an Input stage), which prompt the Desktop user to enter data required to complete the process. All published processes stored in the Blue Prism database can be made available using role permissions to connected Desktop users, and access can be controlled using existing permissions functionality.

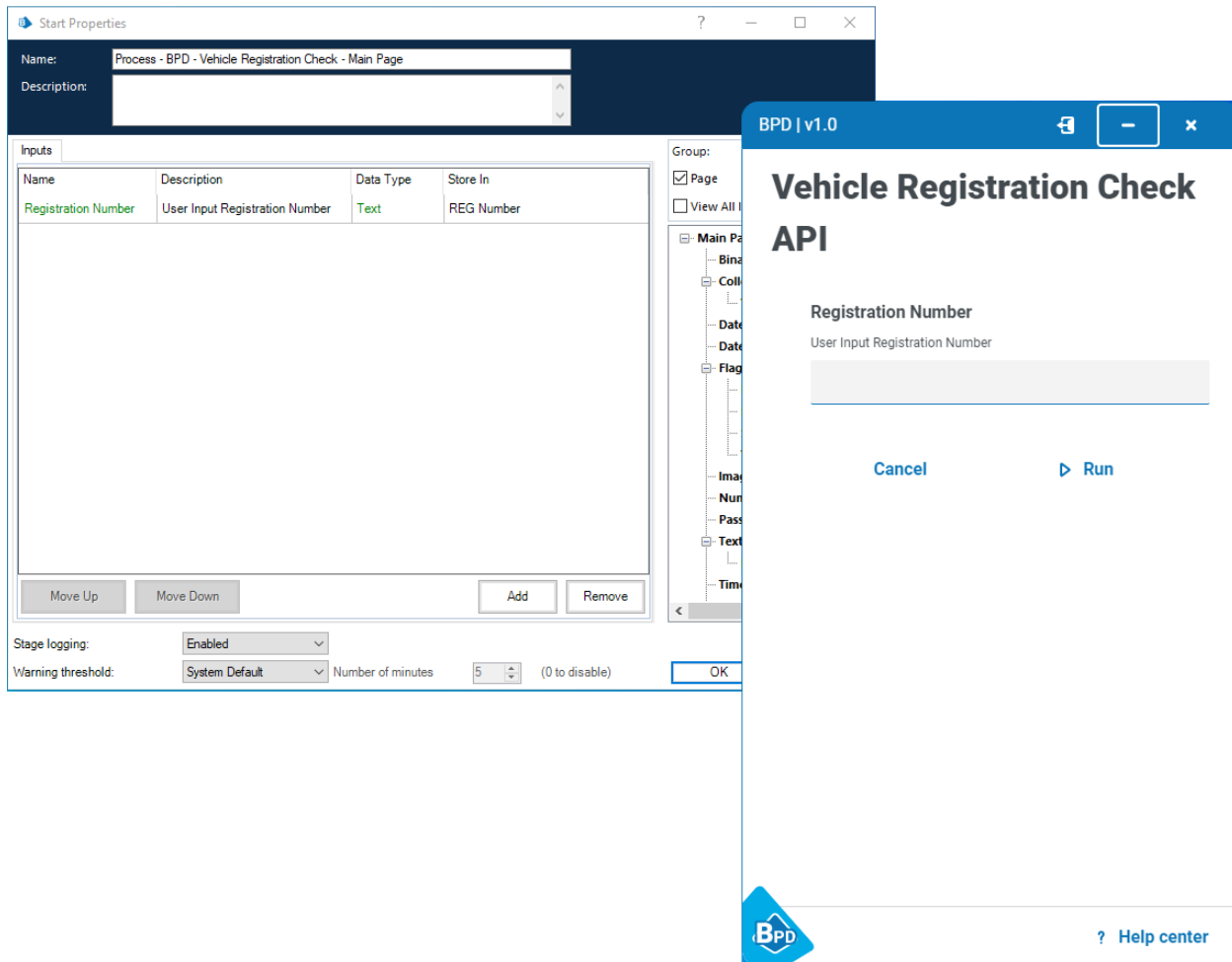
The following design elements should be considered when creating a process to run in Desktop:

- [Start stage](#)
- [Input stage](#)
- [Output stage](#)
- [Enterprise Session stage](#)
- [Session timeout](#)
- [Windows Credentials business object](#)
- [User input blocking](#)

Start stage

The Start stage can be used in Blue Prism processes to add start-up parameters to a process. When input parameters are added to a Start stage, they are displayed as a form in Desktop to enable the Desktopclient user to enter required parameters before running the process. See [Start stage](#) for details.

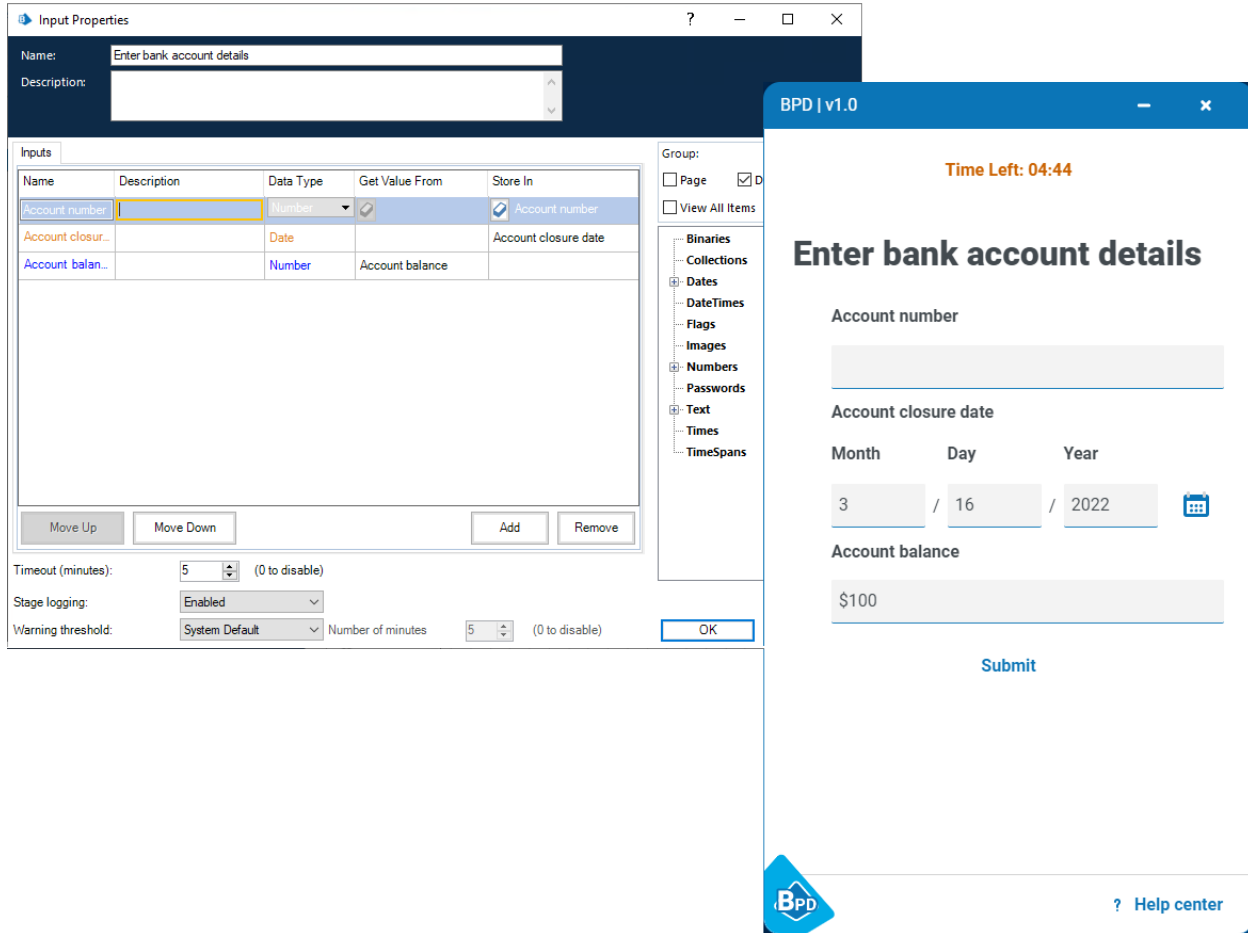
The following image shows the Start stage in Blue Prism and the corresponding form in Desktop.



Input stage

The Input stage can be used in Blue Prism processes to add a form to a process. This stage is only available for users with a Desktop license. See [Input stage](#) for details.

The following image shows the Input stage in Blue Prism and the corresponding form in Desktop.



If your process includes an Input stage, a debug form displays the input fields when run in Process Studio in Desktop debug mode. This enables you to create and test a process without having to publish and run it in Desktop.

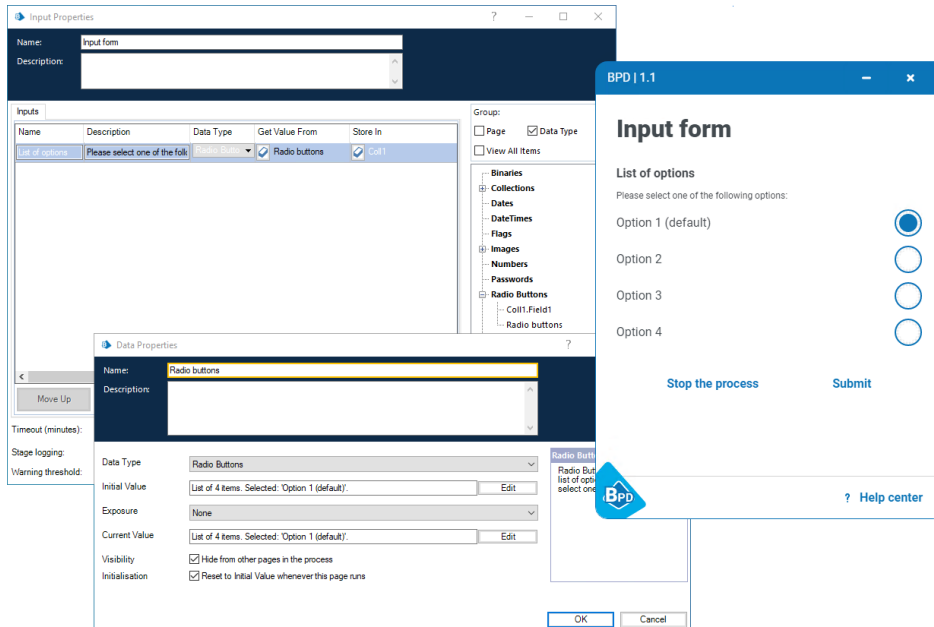



The style of the debug form displayed in Process Studio does not match the style used in the Desktop user interface, but provides an accurate representation of the content.

Add radio buttons to an Input stage

The **RadioButtons** data type can be used in an Input stage to display a list of options to the Desktop user, from which they can select one option. The selected option is then submitted to the process when the user clicks **Submit** on the Input stage form.

The following image shows an Input stage configured with a RadioButtons data type in Process Studio, and the same Input stage displayed in Desktop:

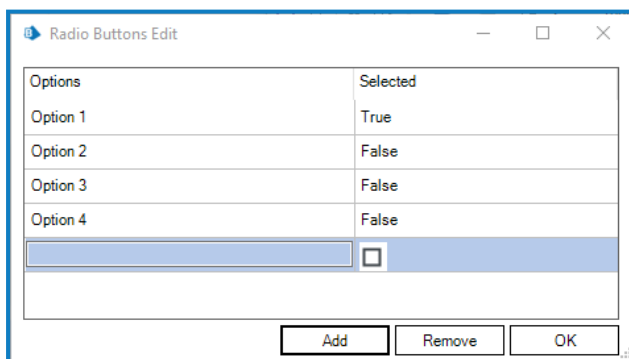


 The **RadioButtons** data type is only available in Blue Prism environments with an active Desktop license applied, and cannot be included in published Blue Prism Enterprise processes.

To make a list of radio buttons available for an Input stage:

1. Add a RadioButtons data type to a data item.
2. Click **Edit** in the Initial Value field.

The Radio Buttons Edit window displays.



3. Click **Add** to list all the available options.

The first option you add will automatically be set as the default selection. You can change this by selecting the checkbox for the required default option in the **Selected** column.

4. Click **OK** to add the list and close the window.

The new data item can now be added to the **Get Value From** field in the Input stage properties.

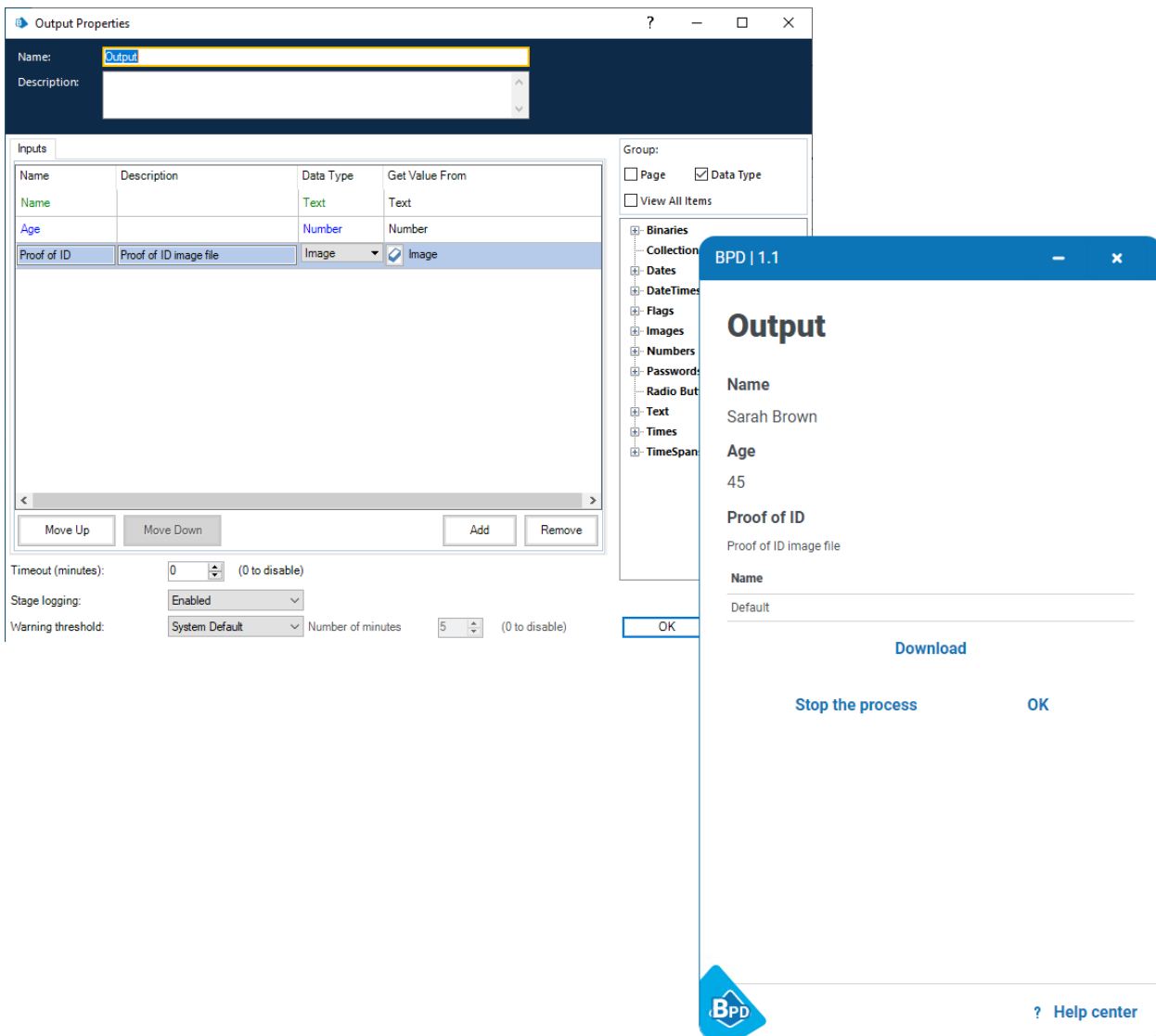
Radio buttons internal business object (IBO)

The Radio buttons IBO includes a **Get selected as text** action, which can be added to the Desktop process to output the selected radio button option as a Text data type. This IBO is only available if Blue Prism Desktop debug mode is enabled in Process Studio.


Output stage

An Output stage can be added to a process to display to the Desktop user a summary of the data that they input during the automation. Output stages can also be used to make any images uploaded via an Input stage available for download by the Desktop user. This stage is only available for users with an active Desktop license. See [Output stage](#) for details.

The following image shows the Output stage in Blue Prism and the corresponding form in Desktop.



If your process includes an Output stage, a debug form displays the input fields when run in Process Studio in Blue Prism Desktop debug mode. This enables you to create and test a process without having to publish and run it in Desktop.

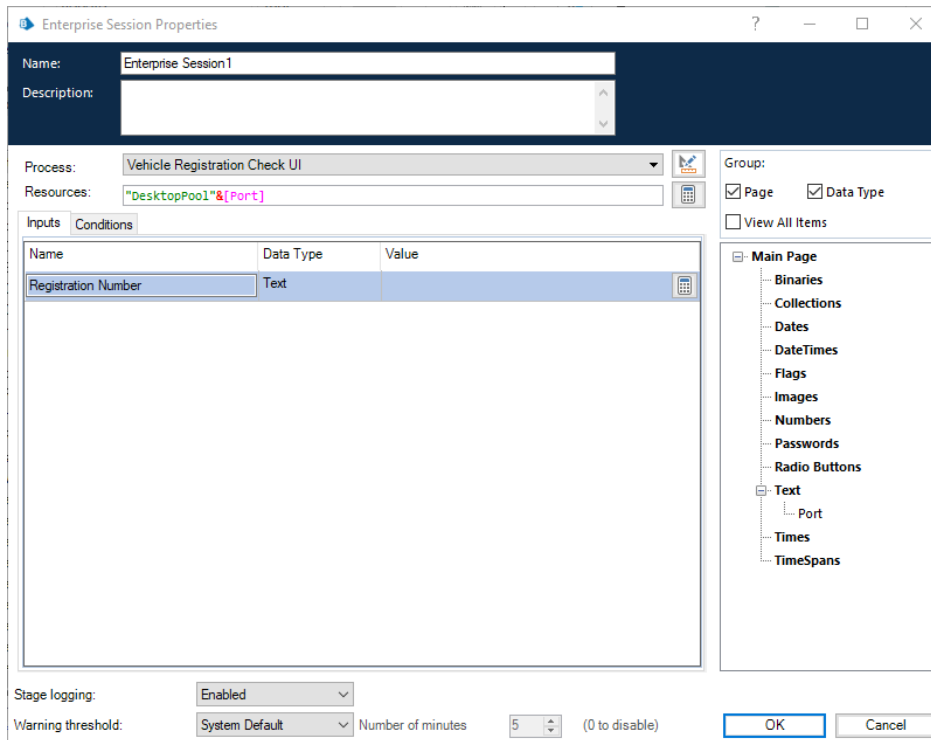
 The style of the debug form displayed in Process Studio does not match the style used in the Desktop user interface, but provides an accurate representation of the content.

Enterprise Session stage

An Enterprise stage can be added to a Desktop process to trigger an automation to run on a specified Blue Prism Enterprise runtime resource, or resource pool. See [Enterprise Session stage](#) for details.

Enterprise Session stage properties

The following image shows the Enterprise Session stage in Process Studio:



Current Enterprise Session stage limitations

When an Enterprise Session stage is successfully run in a Desktop process, a success message displays, indicating that the process has completed. However, this is not an indication that an Enterprise session has been successfully created and started. There is currently no feedback mechanism to the Desktop user once an attempt to trigger an Enterprise session has been made, nor is it possible to schedule a session. It is therefore recommended that resource pools are used where possible to increase the likelihood of a resource being available.

Session logs

By default, once a session has been successfully triggered, a session log is created in Blue Prism Enterprise, which includes the following:

- Inputs passed into the Enterprise process.
- Resource name.
- Enterprise process name.
- Desktop process name.
- Session ID (if generated).
- Session status.

Session timeout

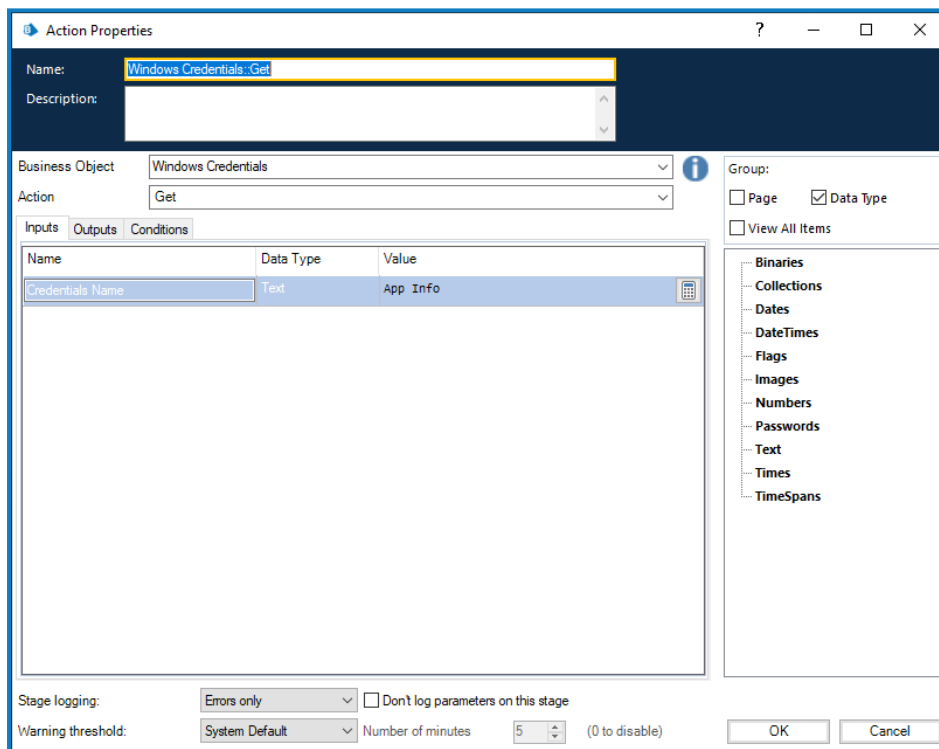
A mandatory 60 minute session limit is automatically applied to all processes running in Desktop, after which time, the process is stopped. If a timeout occurs, it is captured in the session history in the Blue Prism interactive client and displayed on the View issues screen in Desktop.

The 60 minute timer is paused while an input form is open. For details on how to add a time limit (timeout) to a Desktop form, see [Input stage](#). If the **Timeout** setting has not been configured for the Desktop form using the Input stage, then the form will remain open and the 60 minute timer will remain paused. The timer is also paused while an Output stage displays to the Desktop user. For more information, see [Output stage](#).

Windows Credentials business object

As Desktop is a tool used by Desktop users on their machines, using their profile, accounts, and credentials, Desktop processes do not have access to the Blue Prism credential store, and users' credentials are not stored in the Blue Prism database.

The Windows Credentials business object can be used to access users' credentials stored locally for use with Desktop processes. The Windows Credentials business object interacts with credentials stored under Control Panel > User Accounts > Credential Manager > Windows Credentials > Generic Credentials.



The following actions are available:

- **Get** – Retrieves the specified credentials.

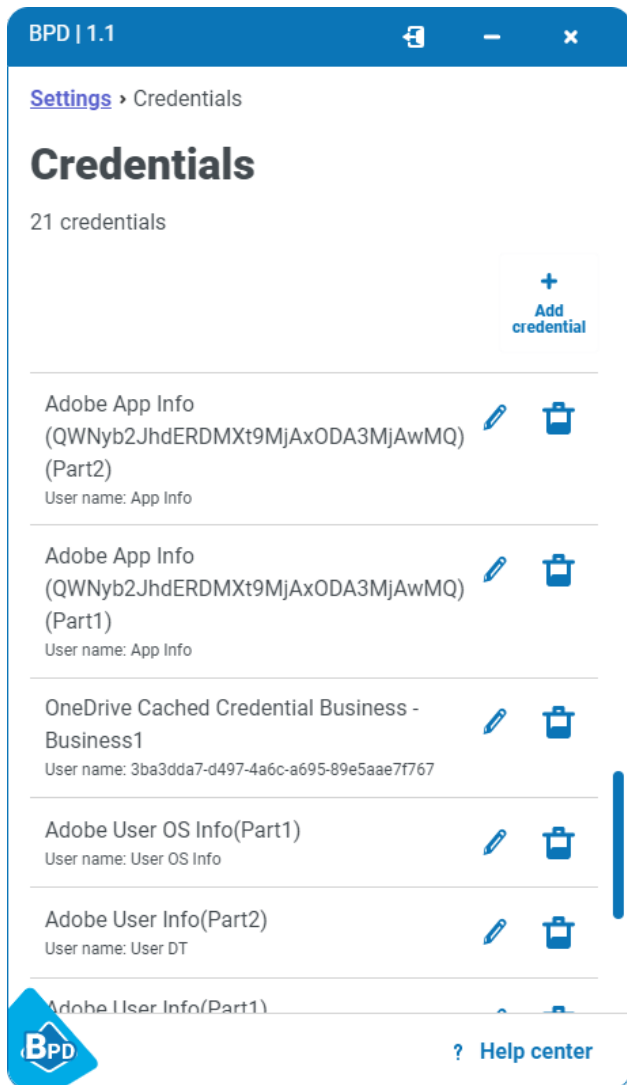
Parameter	Direction	Data type	Description
Credentials Name	In	Text	The internet or network address from the Windows Credential Manager.
Username	Out	Text	The user name for the specified credential.
Password	Out	Password	The password for the specified credential.

- **Set** – Populates given credential name with specified values for username and password.

Parameter	Direction	Data type	Description
Credentials Name	In	Text	The name of the credentials to set. This will be saved in the internet or network address field in the Windows Credentials Manager.
Username	In	Text	The user name for the specified credential.
Password	In	Password	The password for the specified credential.


User credentials in Blue Prism Desktop

In Blue Prism Desktop, users can view, edit, and delete their own credentials from **Settings > Credentials**.



For details see [Configure credentials](#).

User input blocking

 User input blocking in Desktop helps prevent accidental updates by the user while applications are being automated. Generally, users will not be prevented from interacting with other applications while a process is running, however, there are some exceptions which are detailed below.

When Desktop automates an application, user input (for example, mouse clicks and key strokes) is blocked whilst the automation is running:

- If automating a Windows application, only the application being automated is blocked. All other Windows applications on the machine remain operational and will accept user inputs.
- If automating a Java, Microsoft Word, Excel, PowerPoint, Outlook, or browser application, all applications of that type on the machine are blocked. However, only one type of browser window is blocked so, for example, if Chrome is being automated, Edge, or Firefox will continue to work as expected.

Depending on process design, the sending of mouse clicks from Blue Prism may require the target application to be in focus. It is recommended, where possible, that approaches to send mouse clicks to applications that do not require focus are used. Details of these options are listed below:

- Mouse clicks can be sent to Windows applications using the **Press** or **Press Menu Item** actions to avoid moving a user's mouse. When performing a **Press** or **Press Menu Item** action, the Windows application under automation may come into focus. Users can move focus to other applications once the action has completed.
- Mouse clicks can be sent to a Java application in a way that does not move the mouse or steal the focus, as these other actions are all API driven through the Java Access Bridge.
- Mouse clicks can be sent to browser applications using the **Click** action available in Browser spy mode.

Sending key strokes:

- Key strokes can be sent to a Windows application without stealing focus, by using the **Windows Press Keys** action.
- Key strokes can be sent to a target Java application, but not a specific element.
- Text can be written to a browser application using Write stages. There is no option to send individual key strokes to a browser window other than using **Global Send Keys**, which requires the application window to be in focus.

Touchpad actions that are not blocked

When running a process in Desktop, the following touchpad actions related to gestures are not blocked in applications automated by the process:

- Pinch to zoom in and stretch to zoom out.
- Drag two fingers to scroll.
- Tap two fingers to mimic right-click.

Differences in Blue Prism functionality


The following sections describe the differences that apply to the Blue Prism interactive client with a Desktop license applied:

- [Environment functions](#)
- [Business objects](#)
- [Spying](#)
- [Excluded functionality](#)

Environment functions

The *IsStopRequested* environment function is not available in Calculation and Decision stages for business objects and processes. This is because it cannot be used to return a meaningful response in Desktop. Any existing processes that use this function will return *False*.

The *IsBPDEnvironment* environment function can be used in Calculation and Decision stages for business objects and processes to detect whether the process is running on a Desktop runtime resource.

 When running a business object or process that includes the *IsBPDEnvironment* environment function in debug mode in Studio, the function will return *No*, which indicates that a Blue Prism Enterprise resource is running.

The *DesktopSessionTimeRemaining* environment function calculates the number of seconds left in the current Desktop session. The time remaining is displayed to the Desktop user. The timer is paused during Input and Wait stages, as these do not count towards the elapsed time. The output of the function is the time difference in seconds between the maximum session run time limit, and the time elapsed since the session start (excluding time spent during Input and Wait stages).

Internal business objects

The following internal business objects are available for use when using the Blue Prism interactive client with a Desktop license:

- Internal Business Objects – Collections.
- Internal Business Objects – Windows Credentials. See [Windows Credentials business object](#) for details.

No other internal business objects are available with a Desktop license.

Spying

All spy modes available in this release of Blue Prism are fully compatible with Desktop, with the exception of Region mode. Region mode is available, but is not recommended for use in Desktop processes, due to potentially varying screen resolutions and other differing factors.

Excluded functionality

The following Blue Prism Enterprise functionality is not currently available in the Blue Prism interactive client if a Desktop license has been applied:

- **Work queues** – Desktop is not designed to operate like a Blue Prism Enterprise runtime resource, completing one case after another in batch mode, and potentially sharing data with other runtime resources via a common work queue. Desktop is intended to operate in isolation from other runtime resources, controlled by the Desktop user, working cases individually. For this reason, Desktop solutions do not have access to Blue Prism work queues and cannot use the Work Queues VBO. Work queues are not available in System or in Control Room.

Because work queue functionality is not available for Desktop, the **System > Reporting - Work Queue Snapshots** option is also not available.

- **Environment locks** – The Desktop solution does not work simultaneously with other Blue Prism processes, so it has no need for environment locks. The Environment Locking VBO and the Environment Locks screen are not available.
- **Schedules** – Desktop is not designed to work with the Scheduler, therefore schedules are not available in System or Control Room.
- **Calendars** – Desktop is not designed to work with calendars, therefore the Calendars VBO will not function in a Desktop solution. Calendars are not available in System or Control Room.
- **Data Gateways** – Desktop is not designed to work with Data Gateways, therefore the Data Gateways VBO will not function in Desktop solutions.
- **Enterprise credentials** – Desktop client users will use their own local Windows credentials in processes, rather than credentials stored in Blue Prism, so the ability to use credential functionality in Blue Prism is not required.
- **Session control functionality** – The ability to start and stop sessions in Control Room is not available.
- **Automatic archiving** – The ability to select automatic mode on the System - Archiving screen is unavailable.

Desktop process design considerations

This section covers some of the considerations that should be made when creating a solution for Desktop. For details of process design elements which are specific to Desktop, see [Process development for Desktop on page 4](#).

Designing processes for Desktop is slightly different from an unattended RPA solution. With Desktop automations users are directly contributing to the running of a process, with the expectation that the automation runs consistently and successfully. In unattended RPA, workloads are processed in bulk and digital workers loop continuously, working case after case, operating in a clean, consistent, and unobserved environment. By contrast, in Desktop automations, cases tend to come one at a time, the state of the environment is less predictable, and the user is present to observe, participate, and potentially interfere.

It is important that care is taken to ensure that a Desktop solution is designed in a way that offers an optimal user experience, maximizes performance, and mitigates any risk of failure.

Solution design considerations

The following elements must be considered when designing an automation for Desktop.

User requirements

As with unattended RPA, identifying and agreeing requirements and success criteria are key to a successful project. With Desktop automation the Desktop user must also be considered.

Business analysts and solution designers should consider the users as customers, and ideally involve them in the requirements gathering, process walk-throughs, design workshops, and design sign-off. These users are the subject matter experts (SMEs) of the business process and will have a good understanding of the process and its associated workarounds, and problems. This valuable knowledge and experience should be utilized when designing an automation for Desktop.

Having user representation on the project will also help encourage user buy-in and ensure they are informed. Because the ultimate aim is for users to welcome and benefit from automation, their involvement will minimize the chance of low uptake or even rejection of a Desktop solution.

Solution shape

As already mentioned, unattended RPA solutions are meant to work one case after another, so usually utilize loops. Desktop solutions do not involve a work queue and will tend to work one case at a time, as directed by the Desktop user. This means that Desktop processes are more likely to be linear rather than circular in design as loops are not generally used. This isn't to say that a Desktop process could not be designed to repeat certain steps, but by nature, requirements will veer more toward 'one-shot' automation rather than batch work.

Process structure

As with a standard Enterprise solution, your templates should also be used to construct Desktop solutions. This is to assist with time saving, standardization, promotion of best practices, ease of collaboration, and reduction of overall ownership cost.

Preliminary checks

It's very likely that a Desktop solution will need to make multiple initial checks when it starts to run. An unattended automation will operate in a virtual environment built specifically for runtime resources, the machine specification is known and the state of the machine is controllable and predictable.

The Desktop environment is far less clean and more uncertain. Specifications and builds may not be standard, the state of the machine cannot be presumed, and the actions of the user are unpredictable. In summary, a Desktop automation is able to make far fewer assumptions about the environment that it is operating in, and it will need to make more allowances than a Blue Prism Enterprise automation. The following are some of the additional checks that a Desktop automation may need to make:

- Did the user enter valid input parameters?
- What is the state of the target application – is it already running, is the user logged in, what screen is open in the application, or does the application need to be launched?
- Is the version of the target application as expected?
- Is screen resolution important and does the machine meet the requirements?
- Is the Windows version important and does the machine meet the requirements?
- Are the user's credentials in the Windows credential store?

Some form of exception handling will be required to inform the user of the issue where the problem cannot be resolved by the automation.

Minimized applications

It is possible for a Desktop user to minimize an active application from the taskbar while an automation is running. It is therefore recommended that the process design includes checks that the target application is active and visible on screen where required.

Exception handling

For a Blue Prism Enterprise automation, exception handling can be as simple as recording the detail of the problem against the queue item and moving on to the next case. But in Desktop, there is no work queue and there probably is no next case, so the solution designer must make every effort to keep the user informed.

Mid-process terminations should be avoided and instead problems should be reported back to the user gracefully. The language used should be tailored to the user, who may not have technical experience.

The user's actions may be the cause of the problem and the solution designer should attempt to anticipate user interaction. Training will go some way towards controlling user behavior, but the designer of a Desktop automation should expect the unexpected. There is no simple solution to this problem, but being aware of such factors will help when designing processes for Desktop.