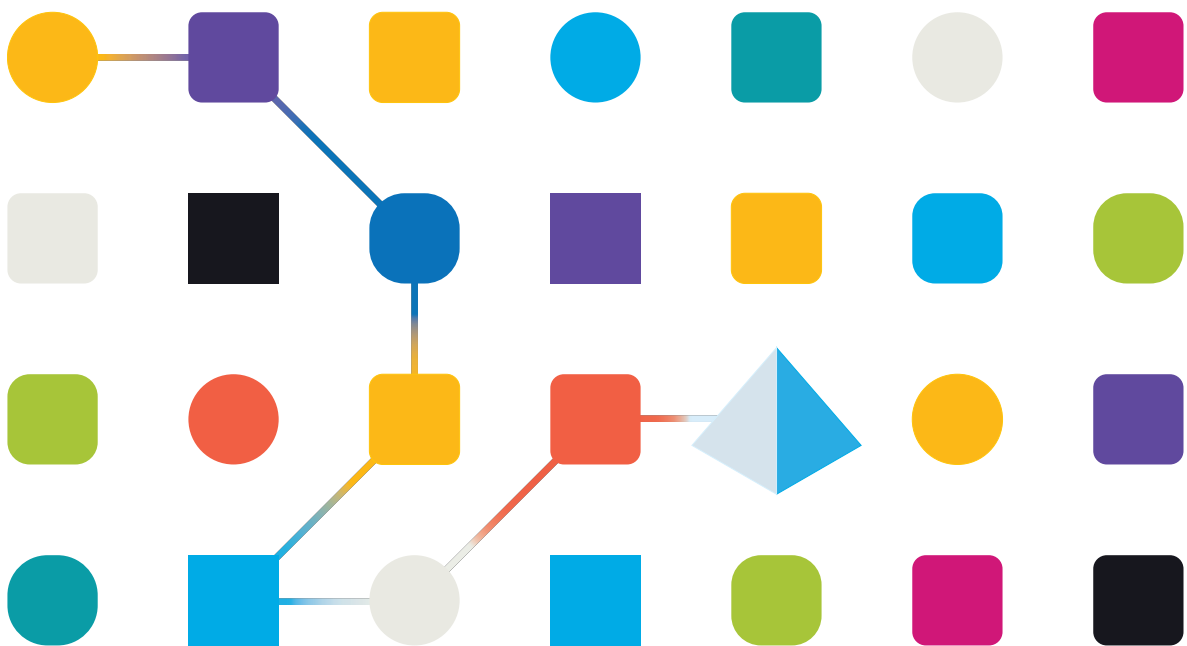




Director 1.0

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

Contents

Director	4
Intended audience	4
Overview	4
Glossary of terms	4
Existing solutions	6
Intelligent Automated Digital Assistant (IADA)	6
Scheduler	6
Orchestration	7
How does Director orchestration work?	7
Using Director orchestration with Interact	8
Using Director with Hub Control Room	8
Using Director	9
Orchestration with Director	9
Director process templates	9
Orchestrator template	11
Collector template	27
Executor template	36

Director

SS&C | Blue Prism Director enables automation developers to align business metrics to their varied workloads, and so use priorities and service level agreements (SLAs) to determine the order in which individual queue items are executed.

Intended audience

This guide is for the SS&C | Blue Prism Blue Prism Cloud 2023.1 platform and assumes that the user has experience in creating automations for SS&C | Blue Prism® Enterprise. It outlines the Director functionality and how to apply orchestration across your workforce and into your automations. This can be applied to both new and existing automations.

It is also advised that users have an in-depth knowledge of work queues and prioritization, and are competent in creating and modifying automations in Blue Prism Enterprise. This guide describes the orchestration templates, and highlights where bespoke customization is required.

Overview

When building process automations, it is easy to fall into the trap of only considering the process design or build at hand. The goal is often to ensure that the process automation runs well, and testing can be focused on this single process automation for any errors or inconsistencies in performance.

What is less often considered, and should be, is how the process automation ecosystem works as a whole. An organization may employ a process controller, or instruct one of the developers to act as a process controller. One of the main responsibilities of the process controller's role is to check on the automations, set up schedules and ensure that work items are being processed based on SLAs, priorities, and demand. This is an inefficient use of resources and adds additional expense.

Using a Web API service and process templates, Director reduces the need for manual scheduling and ensures your most important and urgent tasks are completed first. For details on how to configure the process templates provided with Director, see [Using Director on page 9](#).

Glossary of terms

The following terms are used throughout this guide.

Business priority

The business priority is an integer value above 0 and less than 1000. The lower the value, the higher the business priority. For example, an item with a business priority value of 1 will be processed before an item with a business priority value of 2. The item with the highest priority is advised as the next best item. Where two or more items have the same priority, then the item with the earliest SLA deadline (created time + SLA) will be advised.

Business SLA

The business service level agreement (SLA) value is a time span. This is the amount of time within which the work queue item should be completed and processed. The SLA time span value is used by the Director algorithm to produce an SLA deadline.

Director accepts SLAs in the format DDD.HH:MM:SS. For example, 001.04:30:00 represents 1 day, 4 hours, 30 minutes, and 0 seconds. The minimum SLA value is 000.00:00:01, and the maximum SLA value is 999.23:59:59.

Environment locks

Environment locks provide a locking and token release mechanism, and are essential to enabling orchestration. It is recommended that you are familiar with environment locks and understand how they function. Environment locks are used frequently within orchestration as this concept is key to adding a control mechanism around populating items to the work queues. See [Environment locking](#) for more information.

Existing solutions

Intelligent Automated Digital Assistant (IADA)

Director is an upgrade from IADA. An automated migration path is available for customers currently utilizing IADA.

The IADA tags configured in existing Executor processes will be automatically converted to SLA and priority (as required by Director), as part of your planned migration from IADA to Director. To do this, the migration tool changes the actions in the process templates from IADA to Director's end points using the Blue Prism Director web service business object actions.

Scheduler

The scheduler available in Blue Prism Enterprise is sufficient for small to medium volume implementations of process automations. However, after a period of time you may find that the scheduler does not entirely fit the need of your automations and you may notice some limitations:

- Can I add additional complex logic to trigger schedule runs?
- Can I chain schedules together to remove general down time traditionally seen at the end of a scheduled process run?
- How can I react to changes in priority and SLA across queue items, without manual intervention?
- How can I get more out of my digital workforce?

A continued implementation of the traditional scheduler by itself may lead to a level of inefficiency, which in turn may stop your ability to increase scale without having to add further to your previous investment. For example, add more digital workers when there is still more efficiency to be gained from the workforce.

Depending on your organization's requirements, Director orchestration can be run in addition to Scheduler, but will require a separate pool of digital workers.

Orchestration

In an organization, not all process automations and their associated items have the same business value, some business processes need to be prioritized over others. By default, items will be addressed and processed following a first in, first out (FIFO) methodology. This, combined with the fact that there can be multiple queues that need to be managed, means that work can get stalled when handling an individual queue, or processing automations with work items which are of a lower value. For an automation to be given a priority over another, it has to be configured with a business priority and an SLA, see [Executor template on page 36](#) for details.

Director takes into account business priority and SLA to determine the next best item to run for a digital worker. By default, this would be the item with the highest priority, or if two or more items have the same priority, the item whose SLA deadline (created time + SLA) is the earliest.

How does Director orchestration work?

With Director, orchestration for each digital worker can be controlled at process and/or work queue level. The Orchestrator process will run 24 hours a day (except for a small, planned automation maintenance window), consuming all or part of your digital workforce, without the need for human process controllers.

Blue Prism Enterprise utilizes orchestration by leveraging the Director Web API service and the provided Director templates. Director has two core priorities:

- Adding items to queues, including business priority and SLA.
- Advising the next best item.

The Orchestrator process should:

- Run on digital workers that are dedicated to orchestration, it is then configured to orchestrate the work queues on the digital workers running the Executor process automations.
- Run and operate the same way across the digital workforce environment.
- Continuously reassess the 'Advise Next Best Item' output for all work queues. This also includes items that can be added into the queues on an ad-hoc basis from users using the SS&C | Blue Prism® Interact interface.

With Director, there is an option to decide which process should run on which digital worker, which means it may not be necessary to build all digital workers identically. Different digital workers can have different access privileges and applications installed, depending on the processes that will be executed on that digital worker. Though the capabilities of each digital worker differ, it is recommended that where possible they are equal in what they are able to do. This allows more flexibility on scaling up the resources available for each task.

Using Director orchestration with Interact

In addition to using a Collector process, work queue items can also be fed into work queues through SS&C | Blue Prism® Interact. This enables orchestration to be utilized for human and digital worker collaborations.

SS&C | Blue Prism® Hub administrators can configure Interact forms to add work queue items to target queues on Blue Prism digital workers, and return responses to the Interact user for further input. SLA and priority for the items added from an Interact form are defined when creating or importing the Interact form. The Interact form name is the associated process name that gets added to the queue along with other details. If required, SLA and priority can be overridden or adapted based on the data contained in the Interact form. This can be done through the Interact rules functionality, for example:

IF Job Level = C-Suite THEN Priority = 1.

For details on how to configure an Interact form to use Director orchestration, see [Creating Interact forms](#).

Using Director with Hub Control Room

The Control Room Work Queues page provides you with a summary of all the work queues that are currently running or paused for your connected Blue Prism environments.

The following columns support Director users with performance metrics and SLA reporting. Users can filter and sort data using these categories:

- **Process name** – The name of the process that should be executed against this queue item.
- **SLA** – The service level agreement (SLA) for the work queue item. This is displayed in the format DD:HH:MM:SS.
- **SLA deadline** – The date and time by which the item must be completed in order to meet the SLA. It is displayed in the format DD:MM:YYYY HH:MM:SS. If a work queue item is processed before this time, it is within the SLA.
- **Within SLA** – Indicates if the work queue item was completed within the SLA.
- **Director suggested** – If Director has recommended the queue item as the next best item to process.

You can also use Hub Control Room's sorting and filtering capabilities to produce views that help quantify the success of orchestration. For example, the data provided may allow you to understand:

- Whether only certain digital workers fail to meet SLAs.
- Do specific digital workers take longer on certain work queue items.
- Are some SLAs only met on certain days.

For more information, see [Control Room – Work Queues](#).

Using Director

Director utilizes orchestration in Enterprise by leveraging the Director Web API service within the Blue Prism Director - Orchestrator, Collector, and Executor processes. The Orchestrator process will perform all the logic and intelligent decision making on determining which Executor process to run and when to run that process. The orchestration setup then uses the Collector process and/or Interact to load data from files, applications, and web pages, into the specified work queue, with the defined business priority and SLA. The items are then worked on by the relevant Executor process, as determined by the Director advise next best item endpoint.

Orchestration with Director

By leveraging the Blue Prism Director Web API service and Orchestrator process, you can utilize your Blue Prism Cloud digital workforce in the most effective way possible using the following sequence:

1. Work queues are populated using Interact, or the Collector processes, which can run on separate digital workers as a scheduled task, or on an ad hoc basis. The Collector process uses the Director **Add to Queue** endpoint to populate the item in the work queue using the SLA, priority, and the Executor process name as the main input fields.
2. The Orchestrator process runs on separate digital workers, and the Director service advises the next best item based on priority, SLA, process, and queue names, while advising the next best item using the Director **Advise Next Best Item** endpoint. The main output of the next best item action is the item ID, process name, and queue name, along with any other specific output fields. Based on this output and the Executor process name, the Orchestrator process follows the choice stage to decide which Executor process is called to process the next best advised item. The parameters *Item ID*, *Queue Name*, and *Launch Required* are passed as start-up parameters to the Executor process.
3. The Executor process then uses this input parameter Item ID to fetch the data from the work queue using the Work Queues business object and the **Get Item by ID** action and performs the required automation steps. The item is marked as either an exception or completed in the work queue. After completing the processing of the item by the Executor process the flow moves back to the Orchestrator process. The Orchestrator process follows the same logic/steps to fetch the next best item. If an item is available it is advised or else the Orchestrator process keeps running and waits for the new items from the work queue. The Orchestrator process stops only if the maintenance window time is reached, or a process controller manually sends a stop request from Control Room.

Director process templates

The following three process templates are provided with Director. They form an integral part of the Director solution, but require configuration to meet your requirements.

- [Process - Director - Orchestrator Template](#) – This process provides the foundation for the Director Web API to advise the next best item to be processed from the work queue, based on the business priority and SLA.
- [Process - Director - Collector Template](#) – Loads the data from files, applications, web pages, or Interact into the specific work queue with a defined business priority and SLA.
- [Process - Director - Executor Template](#) – Contains the set of business automation steps to be performed on the work queue items populated by the Collector process.

Importing the Director process templates

If you do not already have the Director templates available to you in Blue Prism, you will need to import them. The Director templates are available for download from the Digital Exchange.

To import the templates:

1. Log into the Blue Prism interactive client, select **File** and click **Import > Release / Skill**.

The Import Release dialog displays.

2. Click **Browse** and locate the Director process template .bprelease file.

3. Click **Next**.

A progress screen displays.

4. When the import has completed, click **Finish**.

For more information on importing release files into Blue Prism, see the [Blue Prism online documentation](#).

Orchestrator template

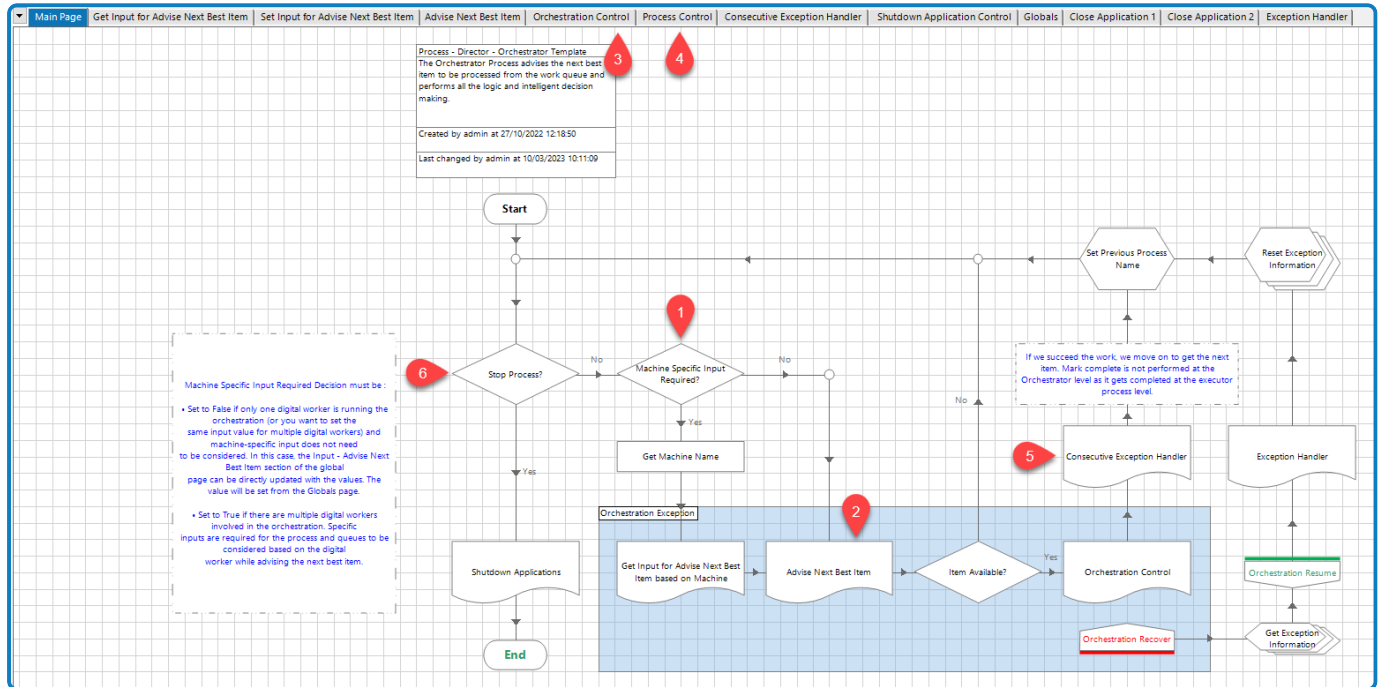
The *Process - Director - Orchestrator Template* process is the core of the orchestration and advises the next item to be processed from the work queue, based on the business priority and SLA. It performs all the logic and intelligent decision making on which Executor process to run and when to run it. It also checks whether an application needs to be launched, or is already launched from the previous run.

This section describes the Orchestrator template, including details of which pages and settings need to be configured for your environment, and which pages and settings can remain static. The following pages are included in the process:

- [Main Page](#) (required) – This is the foundation of the Orchestration process, controlling which Executor process to run and when to run it, based on the item advised by Director.
- [Get Input for Advise Next Best Item](#) (required) – Sets up the input required for the Advise Next Best Item action.
- [Set Input for Advise Next Best Item](#) (required) – Adds the input values to the Process and Queue data items on the global page.
- [Advise Next Best Item](#) (required) – Runs the Director Advise Next Best Item action and returns the next item available for processing, based on the business priority and SLA.
- [Orchestration Control](#) (required) – Determines whether it is necessary to launch an application to process the item, or if the application was launched in the previous item run.
- [Process Control](#) (required) – Determines the Executor process name and executes the required process.
- [Consecutive Exception Handler](#) (optional) – Sends system or internal exceptions thrown by the Executor process as an output to the Orchestrator process.
- [Shutdown Application Control](#) (required) – Closes applications that are not required for the current active process.
- [Globals](#) (required) – The central location for global variables.
- [Close Application](#) (required) – Closes applications opened by the automation.
- [Exception Handler](#) (required) – Records and stores all exceptions. Checks expected or unauthorized exceptions and terminates the process with the appropriate exception message.

Orchestrator template: Main Page

The Main Page is the foundation of the Orchestration process and controls which Executor process to run and when to run it, based on the item advised by Director. This page also determines if the Orchestration process should continue on the digital worker.



The process follows this sequence:

1. The required machine-specific input environment values are checked. See [Environment variables on page 14](#) for details.

Configuration required:

The expression in the *Machine Specific Input Required* Decision stage properties must be updated with one of the following:

- Set to *true* if there are multiple digital workers involved in the orchestration. Specific inputs are required for the process and queues to be considered based on the digital worker while advising the next best item.
- Set to *false* if only one digital worker is running the orchestration (or you want to set the same input value for multiple digital workers) and machine-specific input does not need to be considered. In this case, the Input - Advise Next Best Item section of the global page can be directly updated with the values. The value will be set from the [Globals page](#).

2. Once input values for Advise Next Best Item have been set, the process calls the Advise Next Best Item page and gets the next item for processing.
3. If there is an item available, the process flow moves to the Orchestration Control page. In this page, it is decided if the process to be worked is the same as the process worked for the previous item.
 - If the same process is required, then there is no need to launch the application in the Executor process as the required applications are already launched.
 - If a different process is required, the **Launch required** flag is set to *true* and any previous opened applications are closed.

4. The process flow then moves to the Process Control page where it is determined which Executor process to run, based on the process name derived from the Advise Next Best Item output.

Configuration required:

The Process Control page must be updated when a new process is added/ deleted from the orchestration. In addition:

- Environment variables must be updated for any new queues and processes.
- A new Close application page must be added for any new launched from the new process.

5. Once the Executor process has run, exception details are output from that process. The item is marked as *completed* or *exception* in the Executor process itself. The process flow then moves to the Exception Handler page to check if the same exceptions have been received the specified number of consecutive times. This page can be modified to update the logic for the exception handling based on the business requirement.
6. When the work item completes successfully, the flow then moves on to check if the Orchestration process needs to continue or stop. If it needs to stop, the process closes the application and ends.

Orchestrator template: Get Input for Advise Next Best Item

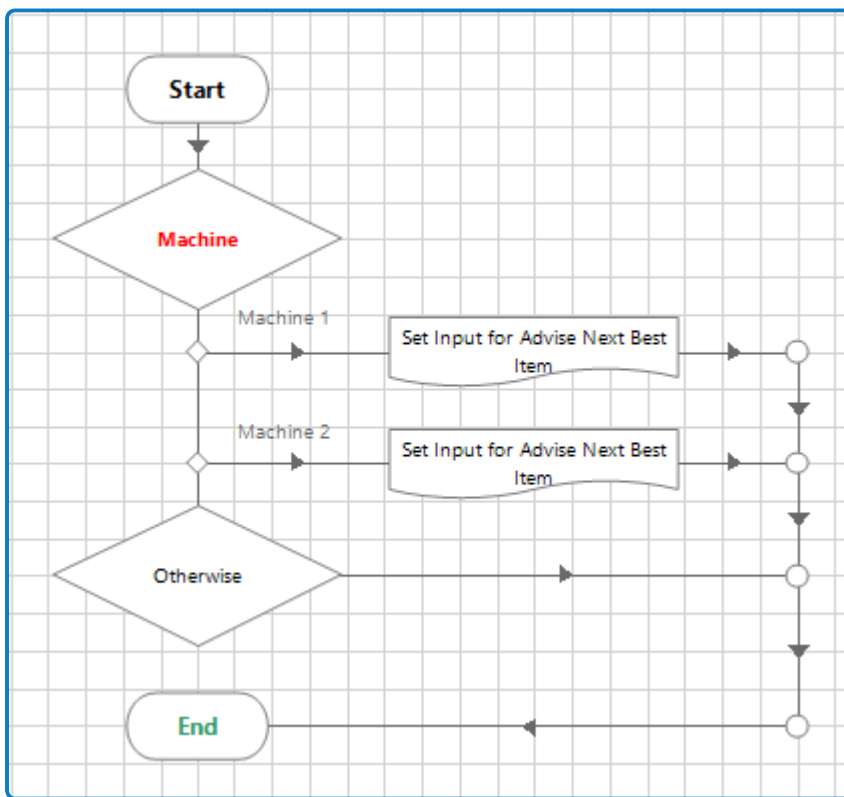
The Get Input for Advise Next Best Item page sets up the input required for the Advise Next Best Item action. The input value is taken from the machine-specific environment variables.

Advise Next Best Item parameters

The Advise Next Best Item action in the Blue Prism Director business object can take the input parameters described below. Based on the machine name on which the Orchestrator process is running, these input parameters can have different values passed to them to decide on the criteria while advising the next best item. See [Orchestrator template: Advise Next Best Item on page 17](#) for details of parameters.

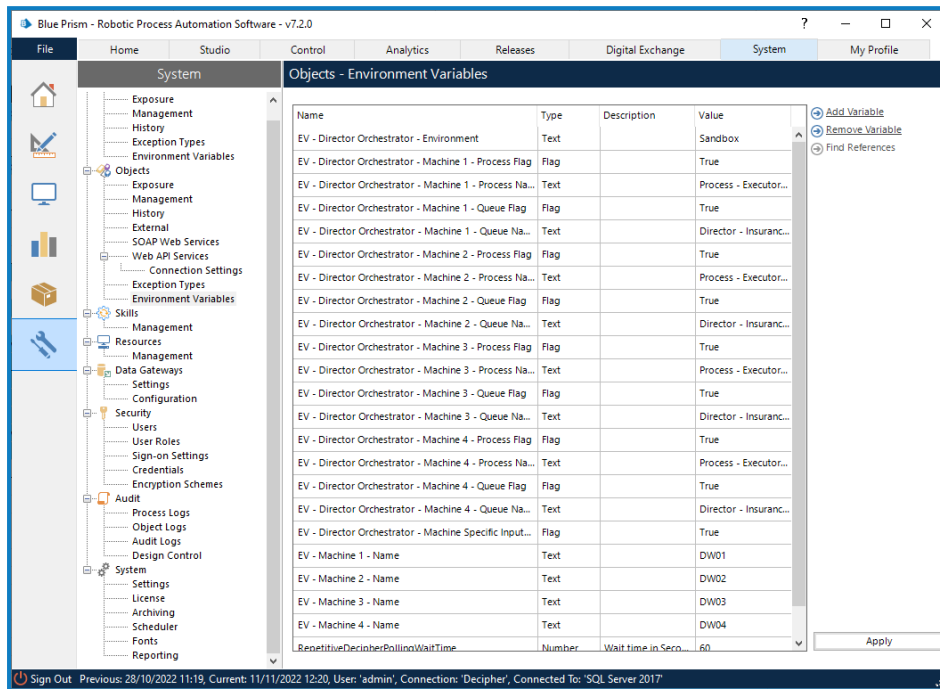
Environment variables

Environment variables for each digital worker available for orchestration need to be created and passed to the choice stage, based on the machine name.



Configuration required:

The Advise Next Best Item parameters all need to be configured using the Environment variables specific to each machine.



The values passed for different machines can vary based on the orchestration requirement.

Machine 1 Environment variables

- [EV - Director Orchestrator - Machine 1 - Process Names] – Comma separated values.
- [EV - Director Orchestrator - Machine 1 - Queue Names] – Comma separated values.
- [EV - Director Orchestrator - Machine 1 - Process Flag] – True or False
- [EV - Director Orchestrator - Machine 1 - Queue Flag] – True or False

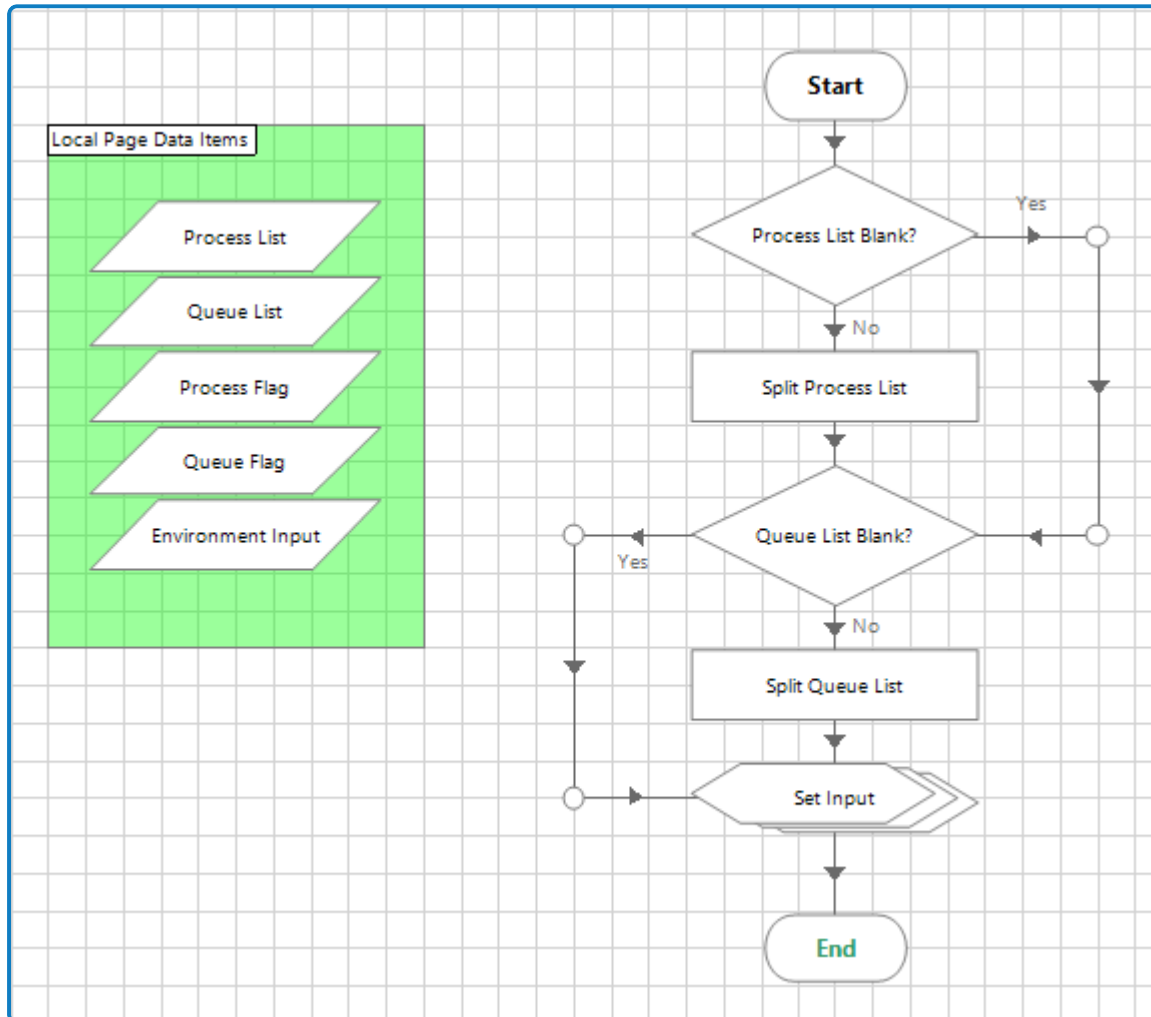
Machine 2 Environment variables

- [EV - Director Orchestrator - Machine 2 - Process Names] – Comma separated values.
- [EV - Director Orchestrator - Machine 2 - Queue Names] – Comma separated values.
- [EV - Director Orchestrator - Machine 2 - Process Flag] – True or False
- [EV - Director Orchestrator - Machine 2 - Queue Flag] – True or False

Orchestrator template: Set Input for Advise Next Best Item

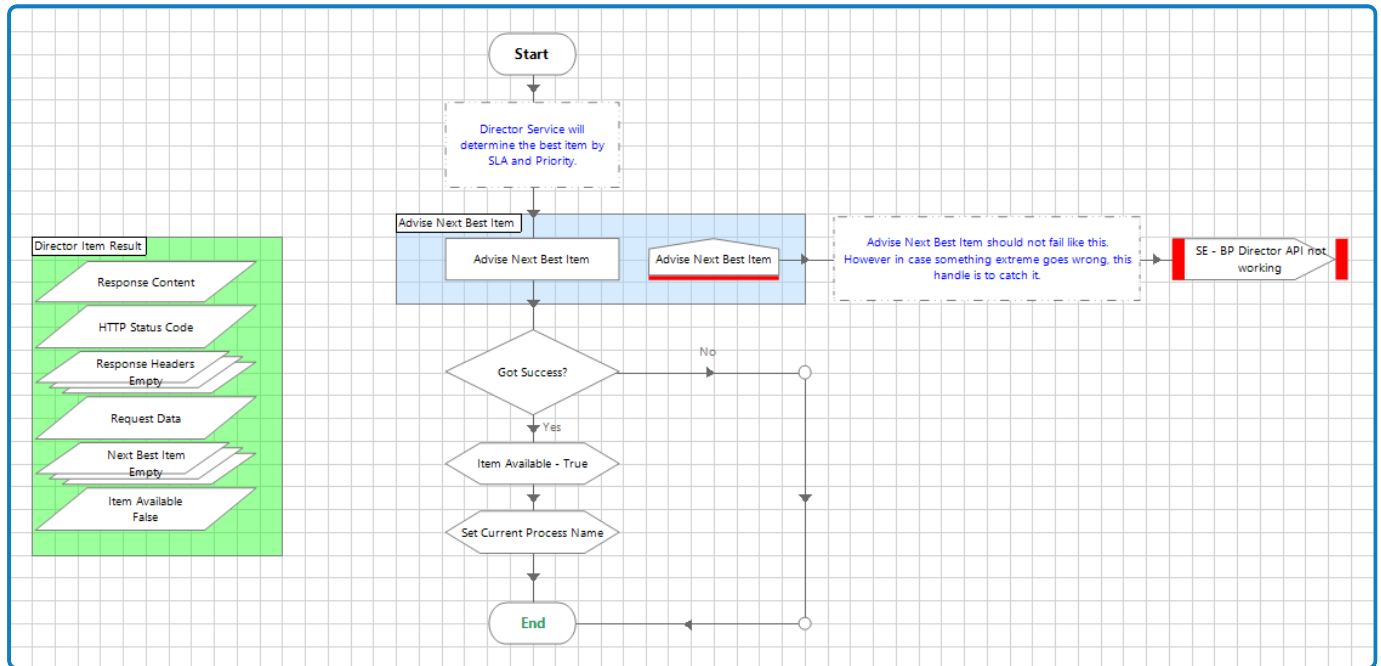
The Set Input for Advise Next Best Item page sets the input values from the machine-specific environment variables to the Process and Queue data items on the Globals page. The values are converted from the comma separated values to the collection data type, as required to populate the Advise Next Best Item input parameters.

The Process flag, Queue flag, and Environment Name are also set from the same page.



Orchestrator template: Advise Next Best Item

The Advise Next Best Item page runs the Blue Prism Director Advise Next Best Item action and returns the next item available for processing, based on the business priority, and SLA.




Director maintains its own view of all the work queues, it then assesses the assigned business priority and SLA to determine and advise the next best item suitable for processing. This could be the item with the highest priority, or if two items have the same priority, the item with the earliest SLA deadline.

Configuration required:

Update the **Advise Next Best Item** action with the following information:

Inputs	Description
Queue Names	Collection containing the names of work queues from which Director will either include or exclude from search scope, see <i>Include Queues</i> .
Include Queues	Flag to indicate whether to include or exclude the queues specified in the Queue Names parameters when searching for the next best item.
Process Names	Collection containing the names of the processes for which Director will either include or exclude from search scope, see <i>Include Processes</i> .
Include Processes	Flag to indicate whether to include or exclude the processes specified in the Process Names parameter when searching for the next best item.
Environment Name	The Blue Prism environment name from Hub. This must match an environment name entered in Hub. This is a mandatory field.

 If the **Include Queues**, **Include Processes**, **Queue Names**, and **Process Names** flag values are all left blank, the next item is advised based only on the priority and the SLA – all queues and processes will be considered in scope when Director performs its search for the next best item to advise.

Action Properties

Name: **Advise Next Best Item**

Description: The Data collection contains the items to be added to the queue. The collection may contain single or multiple rows, with each representing an item to be added.

Business Object: Blue Prism Director

Action: Advise Next Best Item

Group: Page Data Type
 View All Items

Name	Data Type	Value
Queue Names	Collection	[Queue Names]
Include Queues	Flag	[Include Queues]
Process Names	Collection	[Process Names]
Include Processes	Flag	[Include Processes]
Environment Name	Text	[Environment Name]

Stage logging: Errors only Don't log parameters on this stage

Warning threshold: System Default Number of minutes: 5 (0 to disable)

OK Cancel

The following rules are applied, depending on the values returned:

- If **Include Queues** and **Include Processes** flag values are *false* and **Queue Names** and **Process Names** values are blank, the next item is advised based only on the priority and the SLA, and no queue and process values are considered.
- If **Include Queues** and **Include Processes** flag values are *false* and **Queue Names** and **Process Names** have specified values, the next item is advised based on the priority and SLA, but mentioned queue and the Executor process is excluded from the search while advising the next best item. This search is known as a *negative filter*.
- If **Include Queues** and **Include Processes** flag values are *true* and **Queue Names** and **Process Names** have values, the next item is advised only from the mentioned Queues and for the mentioned Executor process based on the priority and SLA. This search is known as a *positive filter*.

HTTP responses from the API

This section shows example responses that are returned from the Director API when the Advise Next Best Item action is executed. For example, `{"adviseNextBestItem":{"processName":"Process - Executor - Insurance Support","queueName":"Insurance Support","itemId":"a16a5ec9-438b-4b7c-b3cd-fe805eb4b1d6"}}`

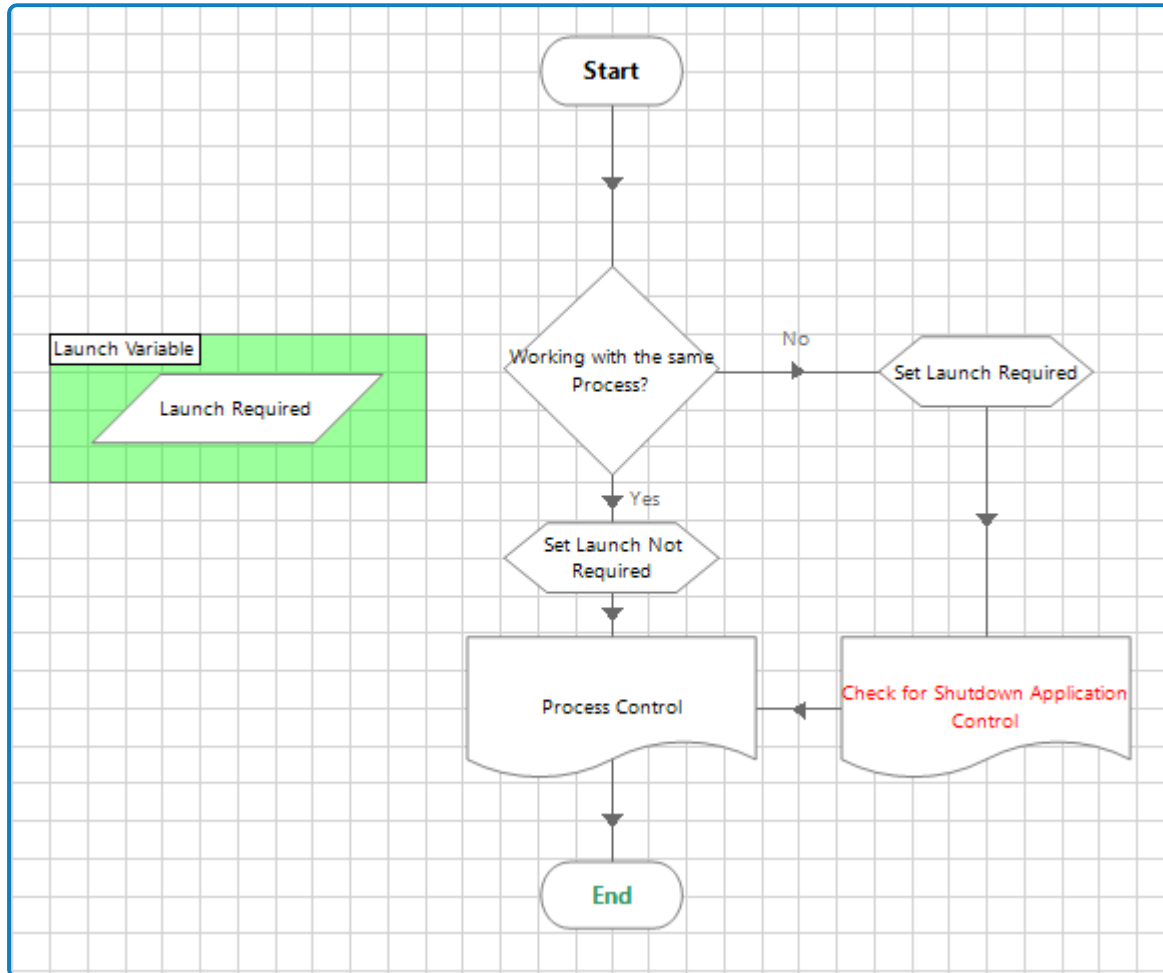
- **HTTP Status Code** – HTTP response status code indicates whether a specific HTTP request has been successfully completed, a status code of 200 indicates success.
- **Response Headers** – HTTP headers that are used in an HTTP response and do not relate to the content of the message, but give a more detailed context of the response.
- **Request Data** – Contains the POST data to API endpoint call
- **Next Best Item** – A collection which consists of the process name, work queue name, and the item ID of the selected work queue item.
 - [Next Best Item.processName] = Process – Executor – Insurance Support
 - [Next Best Item.queueName] = Insurance Support
 - [Next Best Item.itemId] = a16a5ec9-438b-4b7c-b3cd-fe805eb4b1d6

If no item is available for processing based on the input provided for the process and queues, the following values are returned:

- Response Content returns null = `{"adviseNextBestItem":null}`
- HTTP Status Code = 200
- Next Best Item is empty

Orchestrator template: Orchestration Control

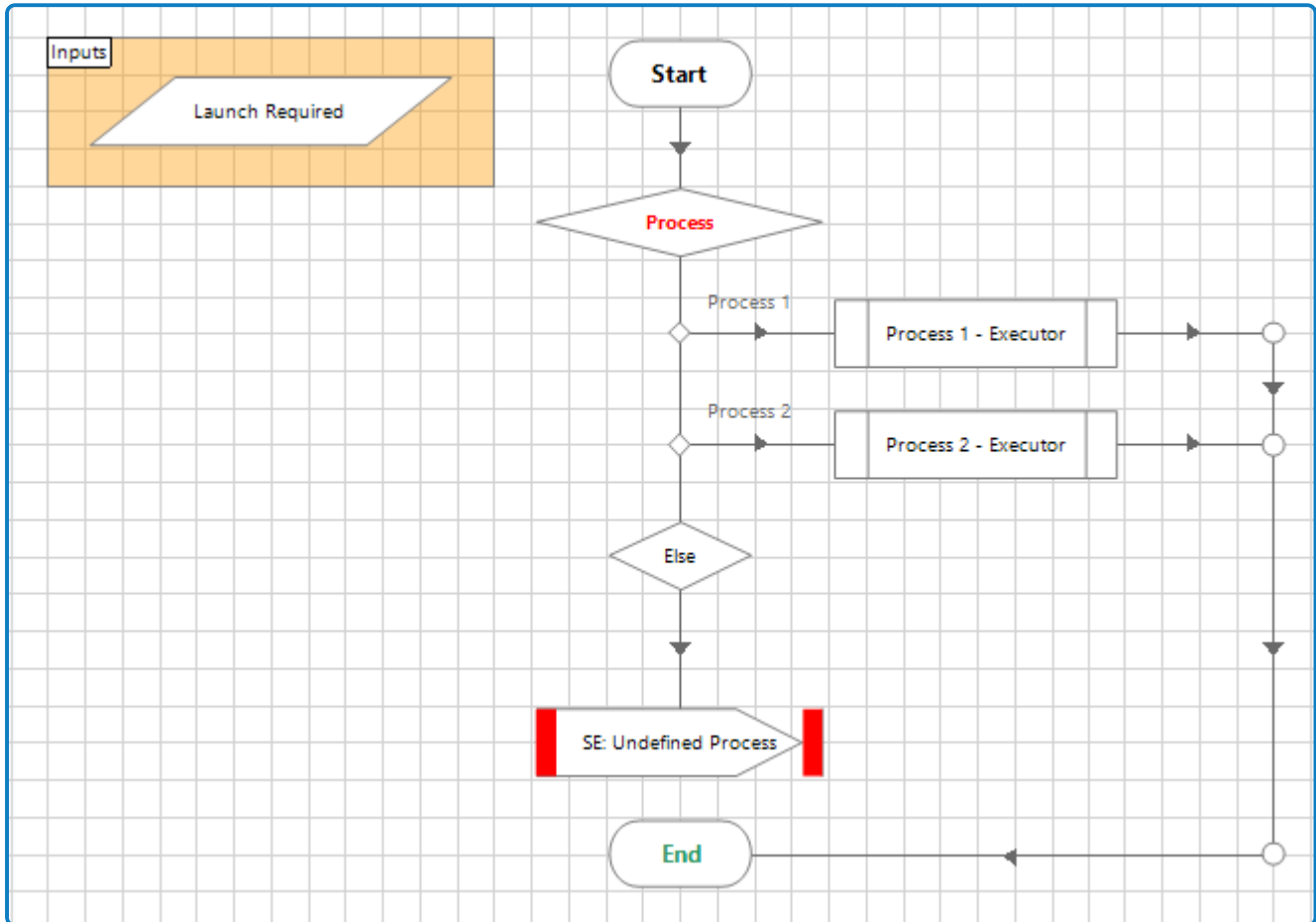
The Orchestration Control page determines whether it is necessary to launch an application. The decision is based on whether the open application is also required by the most recent process to be worked by this digital worker.



- If the required application is already open, control will directly move to the Executor process.
- If the required application needs to be launched, the previously opened application will be closed, and launch will be set to *true* for the Executor process.
- When this *true* flag is passed to the Executor process, the new applications are launched from the Executor process.

Orchestrator template: Process Control

The Process Control page controls which Executor process should be called for processing the item advised by the Director service. The process name, derived from Director's Advise Next Best Item output, is used in the Process Control page to determine which Executor process route should be followed.

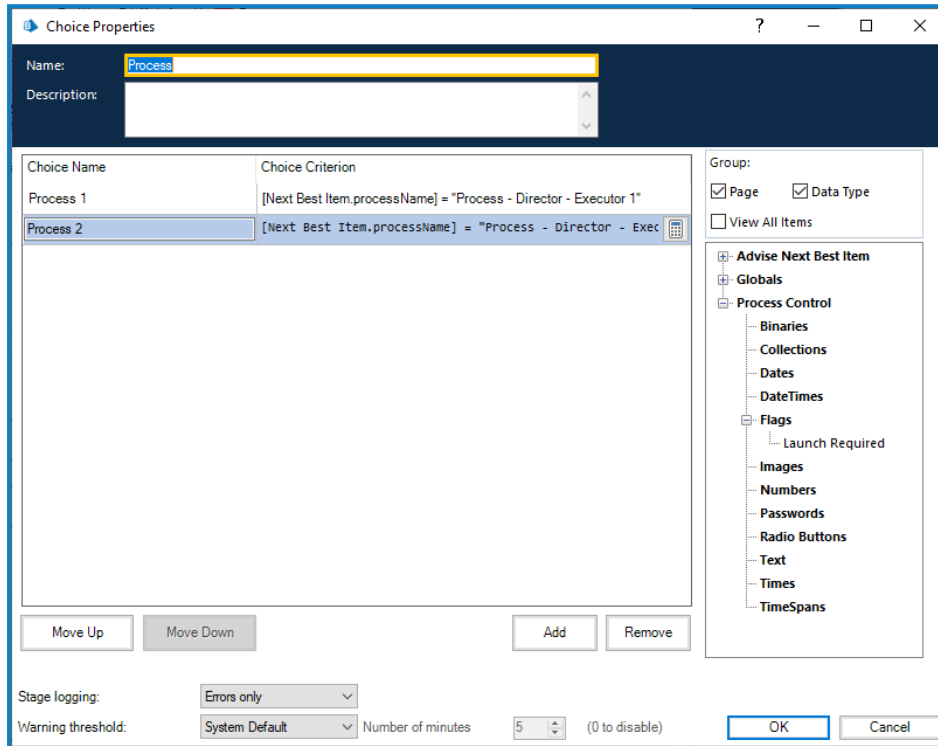


The Executor process name, derived from Director’s Advise Next Best Item output, determines which process route should be followed, using InString logic within the Process Choice stage.

The Launch Required flag, Item ID, and Queue Name input values are passed to the specified Executor process.

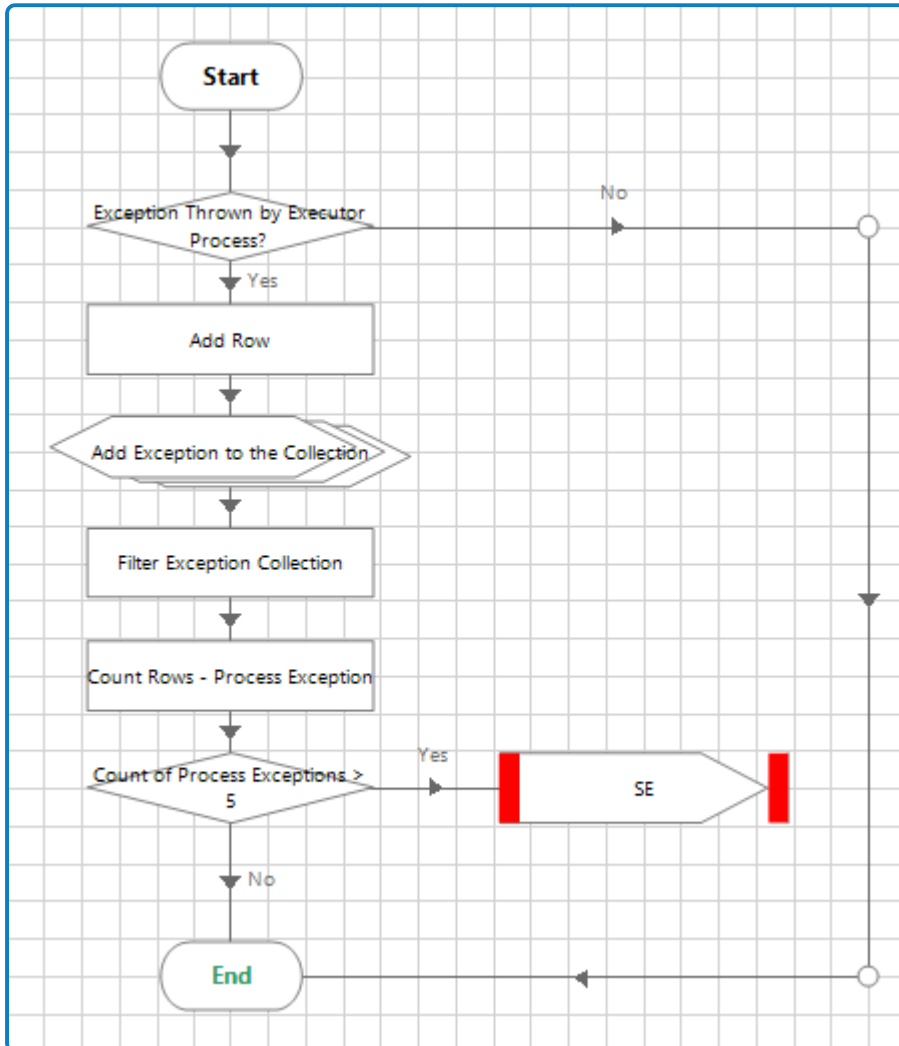
Configuration required:

Update the Choice stage with details of the Executor processes that may be required to run.



Orchestrator template: Consecutive Exception Handler

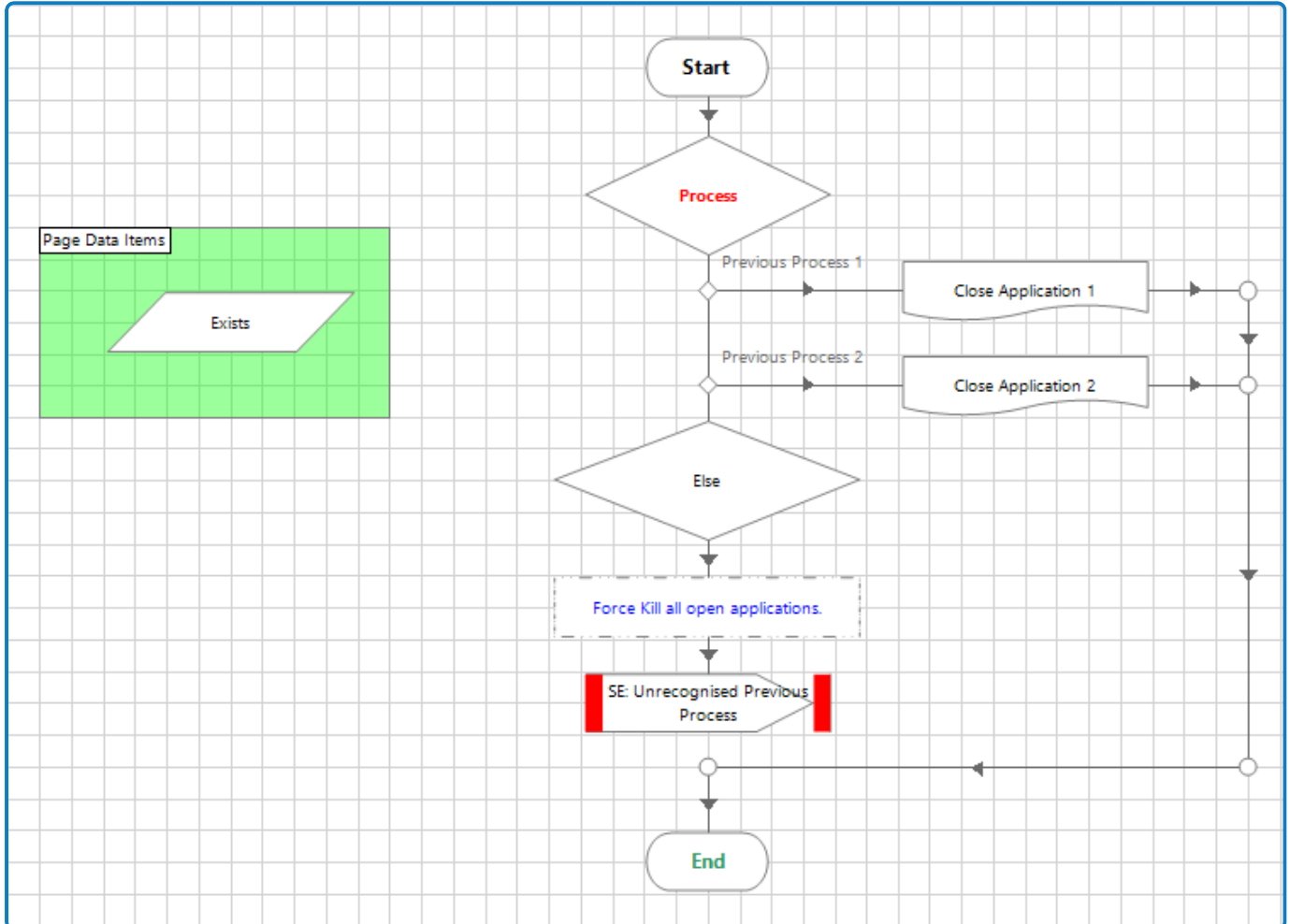
The Consecutive Exception Handler page is optional, and can be designed according to the business needs, which can be used to send system or internal exceptions triggered by the Executor process as an output to the Orchestrator process.



The page checks if any exceptions have been received and if there have been, it adds the exception details and exception type to the Process Exceptions collections along with the Executor process name. It then checks if the same exceptions have occurred consecutively for the same Executor process and the Orchestrator process is terminated.

Orchestrator template: Shutdown Application Control

When it is determined from the Orchestration Control page that the current advised item is to be processed by a different Executor process than the one used for the last run, the flow moves to the Shutdown Application Control page to close down any applications that were opened for the previous item processing.



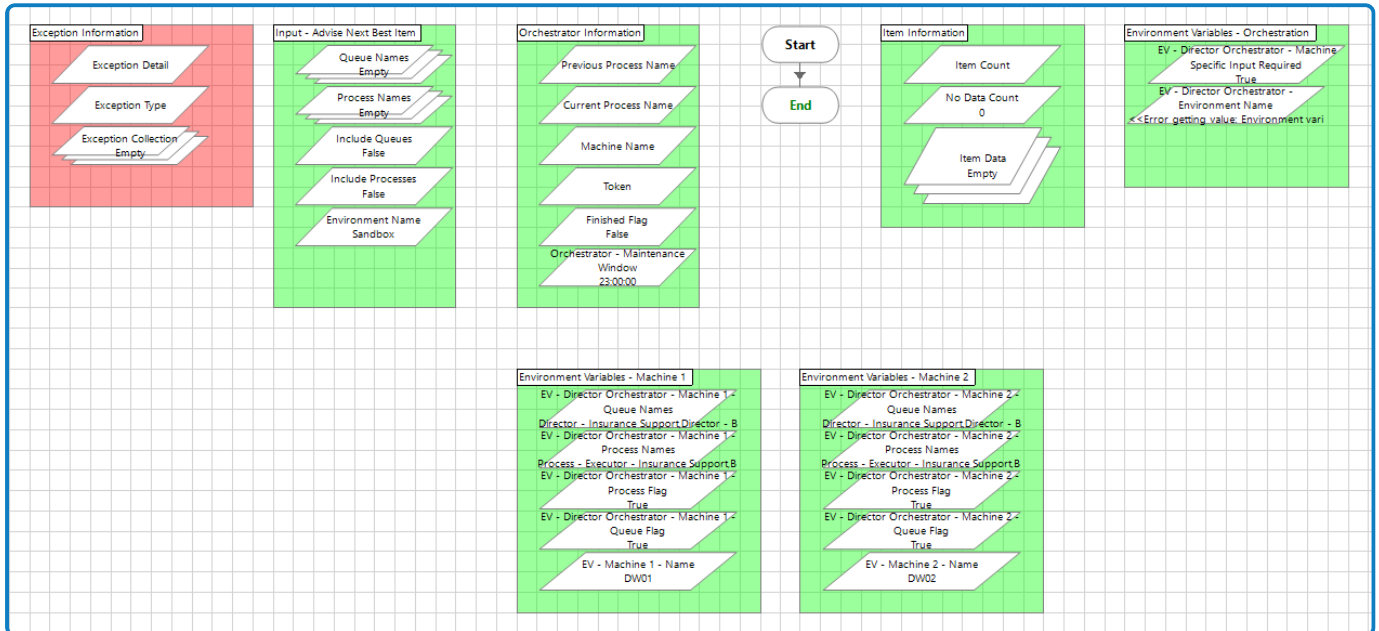
An Executor process will no longer be worked if:

- All of its queue items have been worked.
- A higher priority Director item has been found and the next Executor process is changed to that of the higher priority item.

The applications that were previously open and being used by the process are shut down. This returns the digital worker to its initial state with no applications open.

Orchestrator template: Globals

The Globals page is the central storage location for global variables for the Orchestrator template. This page also allows for a clear distinction on what is global and what is not. Any data item outside of this page should be assumed to not be globally accessible by this process.



Input parameters for the Advise Next Best Item action, all the environment variables, exception data items, and orchestration related parameters, such as machine name, previous process, current process, token, and so on, are available on this page.

Configuration required:

When a process runs, the Set Input for Advise Next Best Item page retrieves the values from the environment variables and populates the following data items:

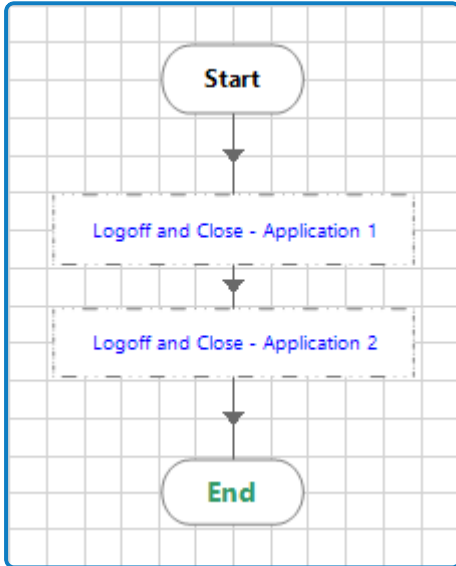
- Queue names
- Process names
- Include Queues flag
- Include Processes flag
- Machine name (digital worker)
- Environment name (the same name is used for all machines and only needs to be set once)

Use the *positive filter* if you want to include the specified processes or queues when Director advises the next best item. This filter excludes those processes or queues that are not added in the process or queue collection. If you choose this filter, every time a new process or queue is added to the orchestration, the environment variables will need to be updated to include the new process or queue, if required.

Use the *negative filter* if you want to exclude the specified processes or queues when Director advises the next best item. This filter includes all those processes or queues that are not added in the process or queue collection. If you choose this filter, every time a new process or queue is added to the orchestration, it will automatically be included without changing the environment variables. If you need any new process or queue to be excluded when Director advises the next best item, you must explicitly add that to process or queue collection by changing the environment variables.

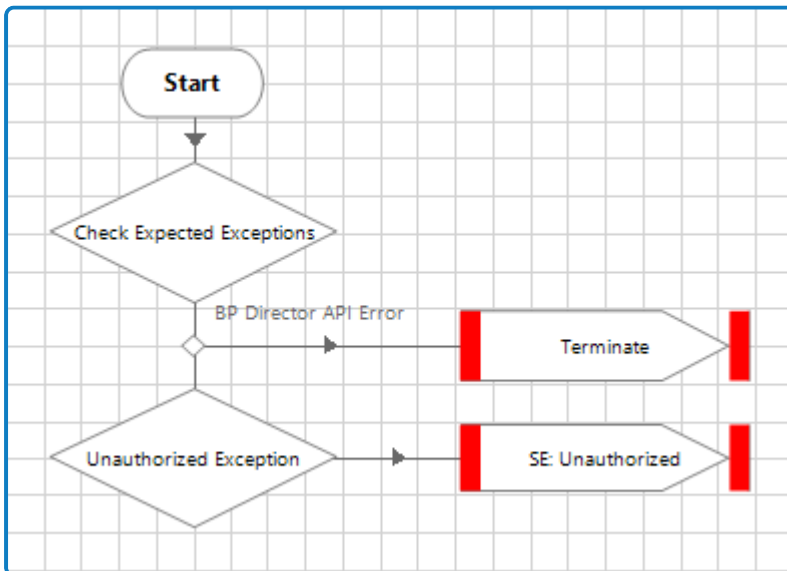
Orchestrator template: Close Application

You must create a Close Application page for each application opened when the automation runs. These pages then need to be called from the Shutdown Application Control page. This page ensures that each application is closed when moving on to the next application, or when a request stop is done.



Orchestrator template: Exception Handler

The Exception Handler page checks for any expected or unauthorized exceptions and terminates the process with the appropriate exception message.



Collector template

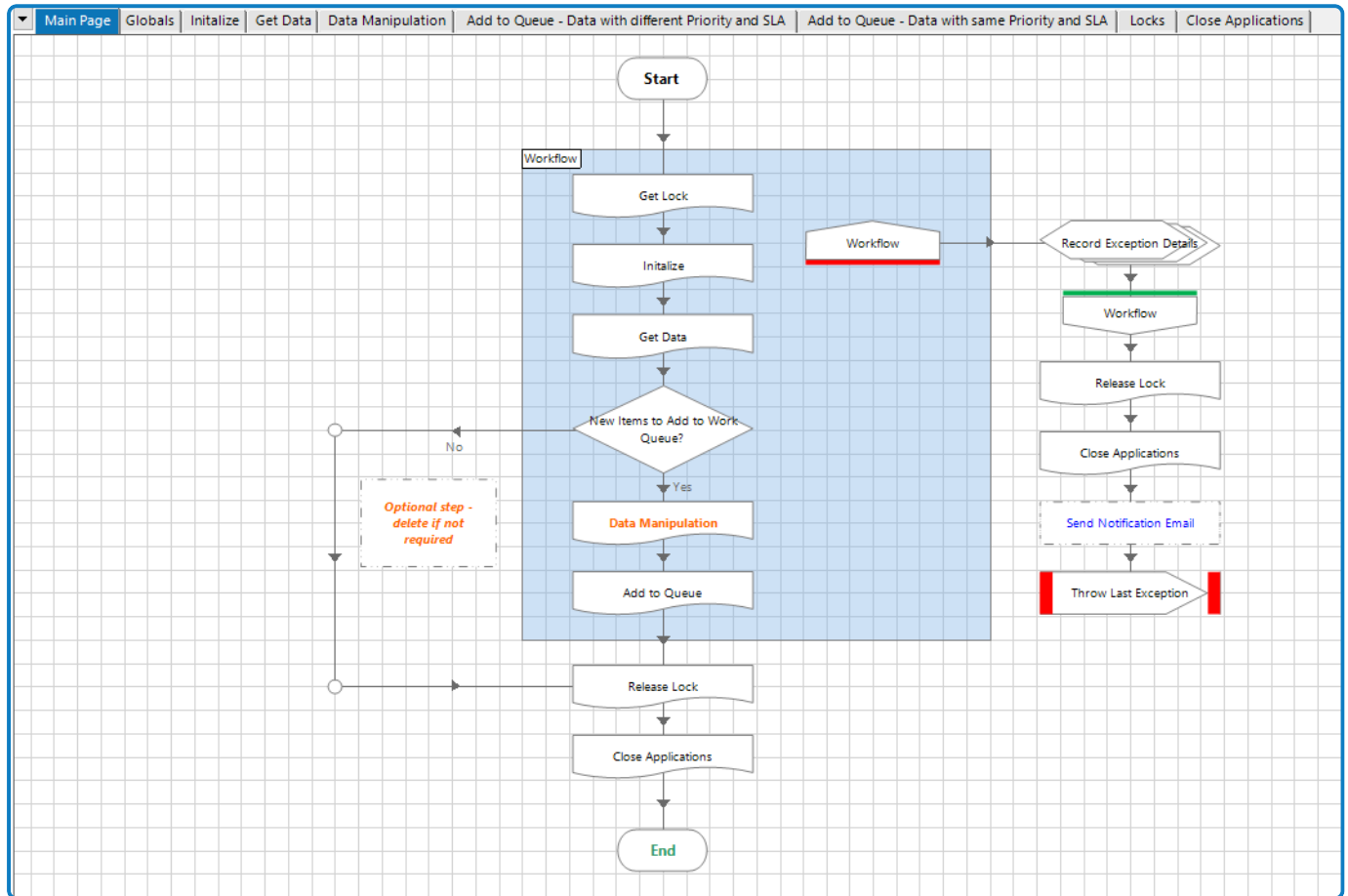
The *Process - Director - Collector Template* loads the data from files, applications, web pages, and so on into the specific work queue with defined business priority and SLA. The items added from the Collector process are worked on by the Executor process. It is recommended that this process is run on a separate machine that is not running the Orchestrator process.

This section describes the Collector template, including details of which pages and settings need to be configured for your environment, and which pages and settings can remain static. The following pages are included in the process:

- [Main Page](#) (required) – The Collector process loads the data from files, applications, web pages, and so on into the specific work queue for the business process with the required priority and SLA.
- [Build Help](#) (optional) – Provides guidance on how to maintain consistency across the process template.
- [Globals](#) (required) – The central location for global variables for this process template.
- [Initialize](#) (required) – Launches the required applications for the Collector process prior to any application interactions to obtain the data needed to load the work queue.
- [Get Data](#) (required) – Handles the work steps/logic to obtain the data to load the queue.
- [Data Manipulation](#) (optional) – Can be used if any work queue items require data manipulation. If not required, this page can be removed.
- [Add to Queue - Data with different priority and SLA](#) (optional) – Uses the Add to Queue action to add the data obtained from the Get Data page and the Data Manipulation page (if applicable) to the specified work queue with the process specific priority and SLA provided. This page should be used if items with different SLAs and priorities are to be added from the Collector process.
- [Add to Queue - Data with same priority and SLA](#) (optional) – Uses the Add to Queue action to add the data obtained from the Get Data page and the Data Manipulation page (if applicable) to the specified work queue with the process specific priority and SLA provided. This page should be used if items with the same priority and SLA are to be added from the Collector process.
- [Locks](#) (required) – Used to handle the environment lock functionality by calling the desired action.
- [Close Applications](#) (required) – Ensures that all applications that are interacted with during the execution of the process are closed.

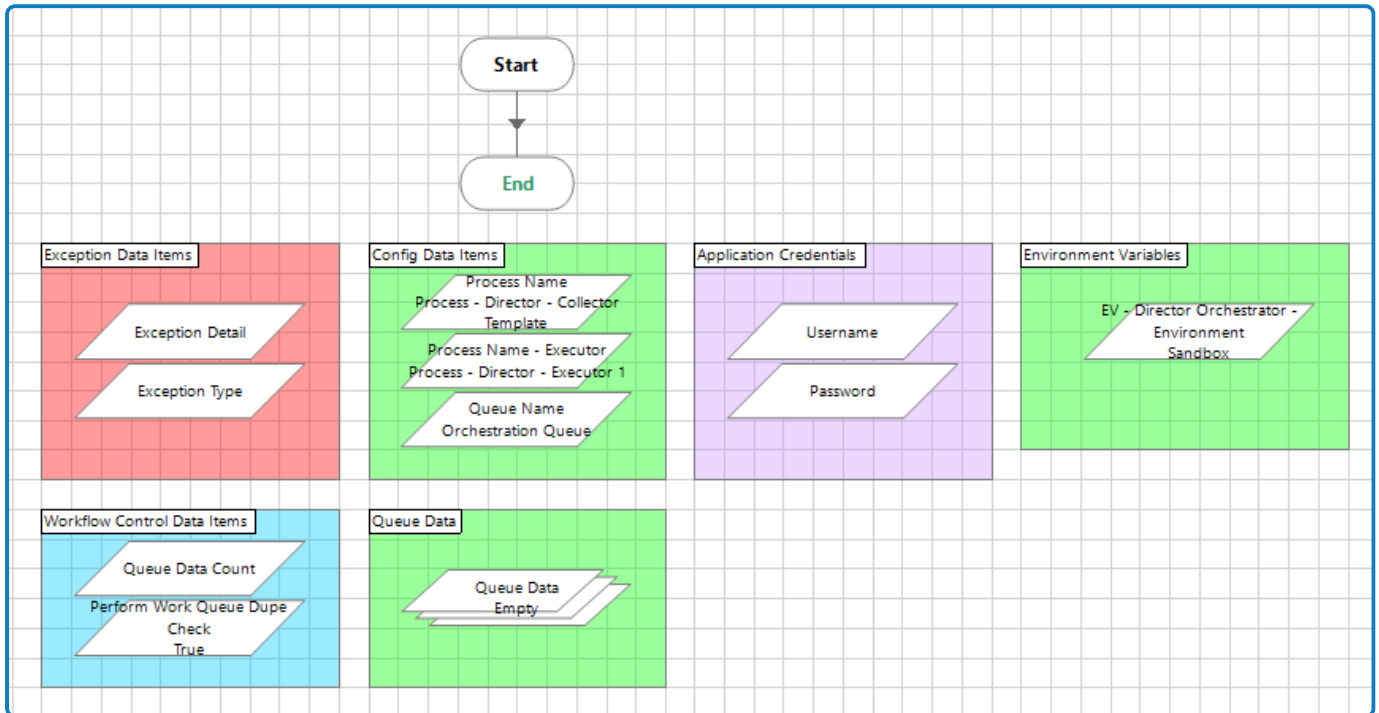
Collector template: Main Page

The Main Page is the foundation of the Collector process and controls exceptions, and the order in which the pages in the template are processed. Each page in this process are detailed below.



Collector template: Globals

The Globals page provides a distinct, central location, for the storage of global variables.



Data Added output

The Data Added parameter is an output parameter of the Collector process. It has an initial value of *false* and is set to *true* after items have been added to the work queue. This data item is passed back up to the layer above the Collector process if any specific logic is required as a result of a successful or unsuccessful queue load.

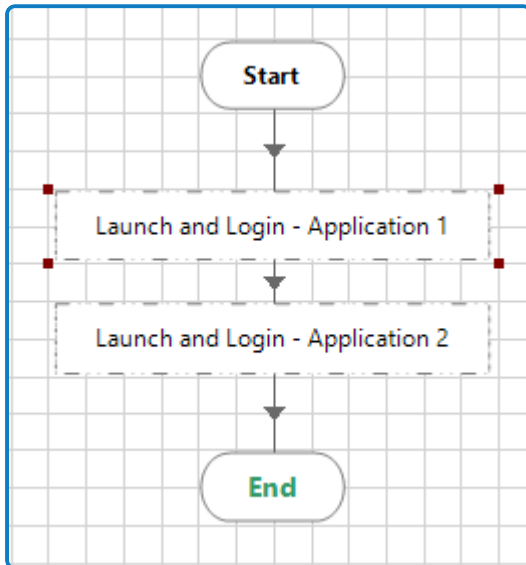
Configuration required:

Update the following data items in the Config Data Items block:

- **Process Name (collector process name)** – This variable is used in numerous places, such as when sending exception emails, and obtaining and releasing the environment locks used on the main page.
- **Process Name (executor process name)** – Used when adding data to the work queue on the Add to Queue page.
- **Queue Name** – Used when adding data to the work queue on the Add to Queue page.

Collector template: Initialize

The Initialize page launches the applications for the Collector process and retrieves any data needed to load the work queue, before interacting with any applications.



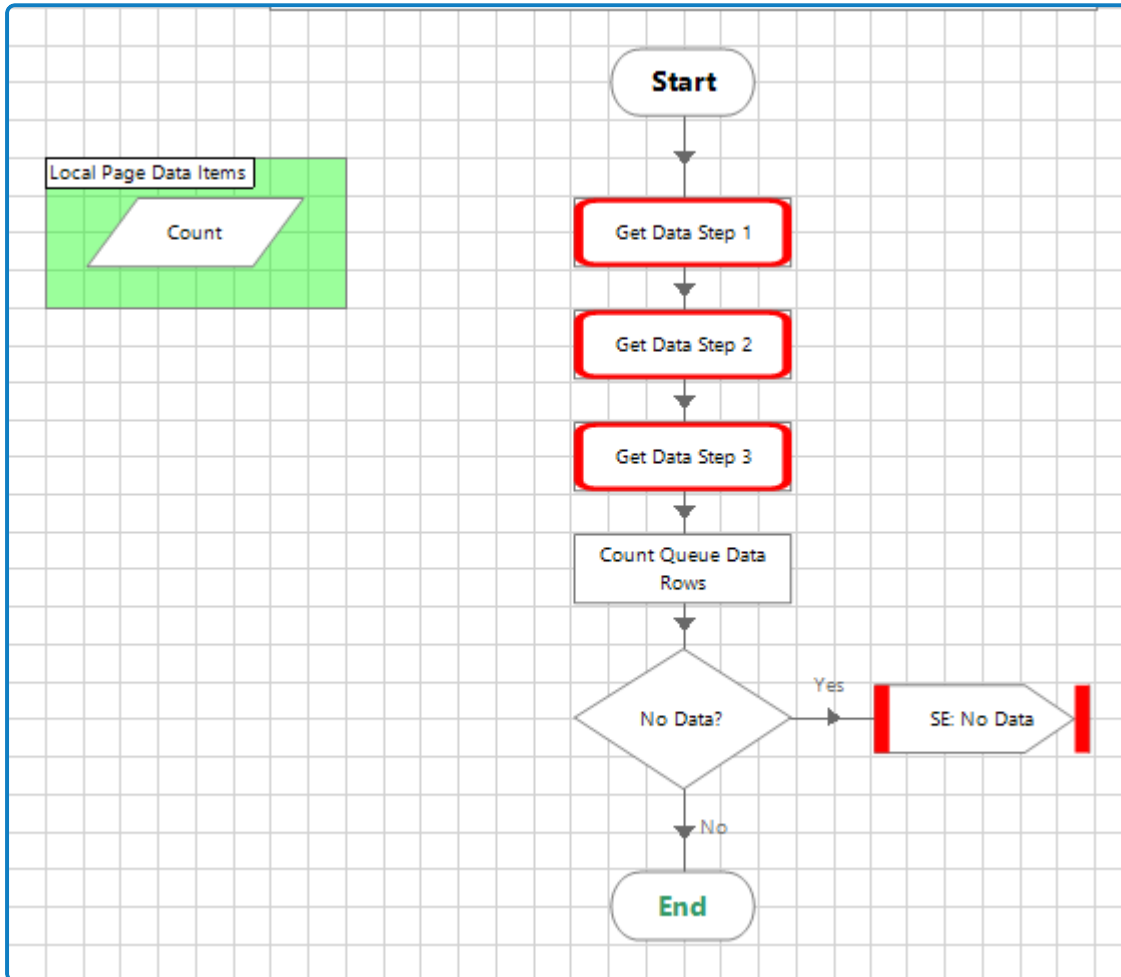
Depending on the application and configuration, these steps can vary from a simple launch of an application, up to a number of actions that need to be performed, such as launching an application, selecting a sub-system, retrieving user credentials for the sub-system, inputting credentials, and selecting a login option.

Configuration required:

Add the necessary stages or pages to launch and log into all applications that the Collector process will interact with.

Collector template: Get Data


The Get Data page handles the work steps/logic to obtain the data to load the queue. The data is then ready to be manipulated if required, before being added to the work queue. If there is no data to be added after completion of the work steps, an exception is triggered to provide notification.



Configuration required:

Add any steps required to retrieve the required data for the work queue, such as:

- Navigate to any screens from which data needs to be retrieved.
- Add actions to open files and retrieve required data.
- Update the Get Data step actions to add data to the collection.

 Data items relating to the **Queue Data** collection are stored in the Globals page.

Collector template: Data Manipulation

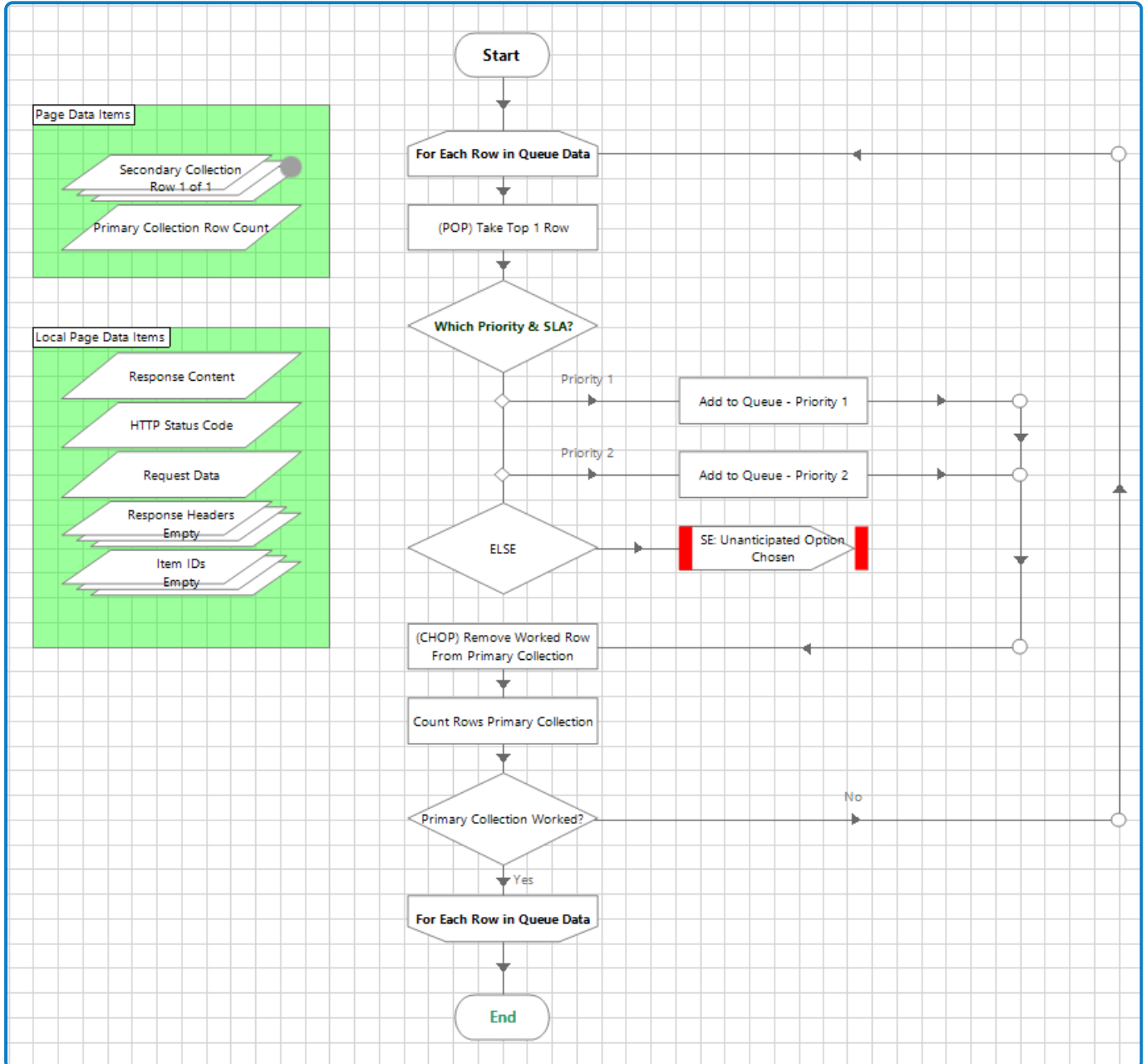
The Data Manipulation page is an optional page to be used if any data manipulation is required before being added as work queue items. If not required, please remove the page reference from the main page.

Configuration required:

Add the necessary stages or pages required to manipulate data before the item is added to the work queue.

Collector template: Add to Queue - Data with different priority and SLA

The Add to Queue - Data with different priority and SLA page uses the Blue Prism Director Add to Queue action to take into account any process-specific priority and SLA to add the collection of data to the specified work queue. Use this page to add data to the work queue when each row in a **Queue Data** collection needs to be assigned a different priority and SLA.

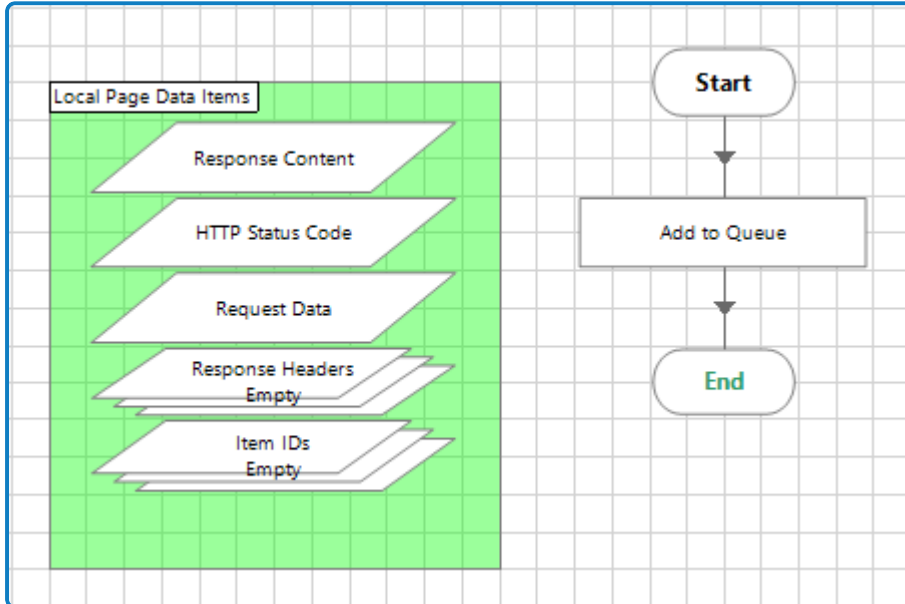


Configuration required:

Update the **Add to Queue** action inputs to match your requirements.

Collector template: Add to Queue - Data with same priority and SLA

The Add to Queue - Data with same priority and SLA page uses the Director **Add to Queue** action to take into account any process-specific priority and SLA to add the collection of data to the specified work queue. Use this page to add data to the work queue when each row in a **Queue Data** collection needs to be assigned the same priority and SLA.

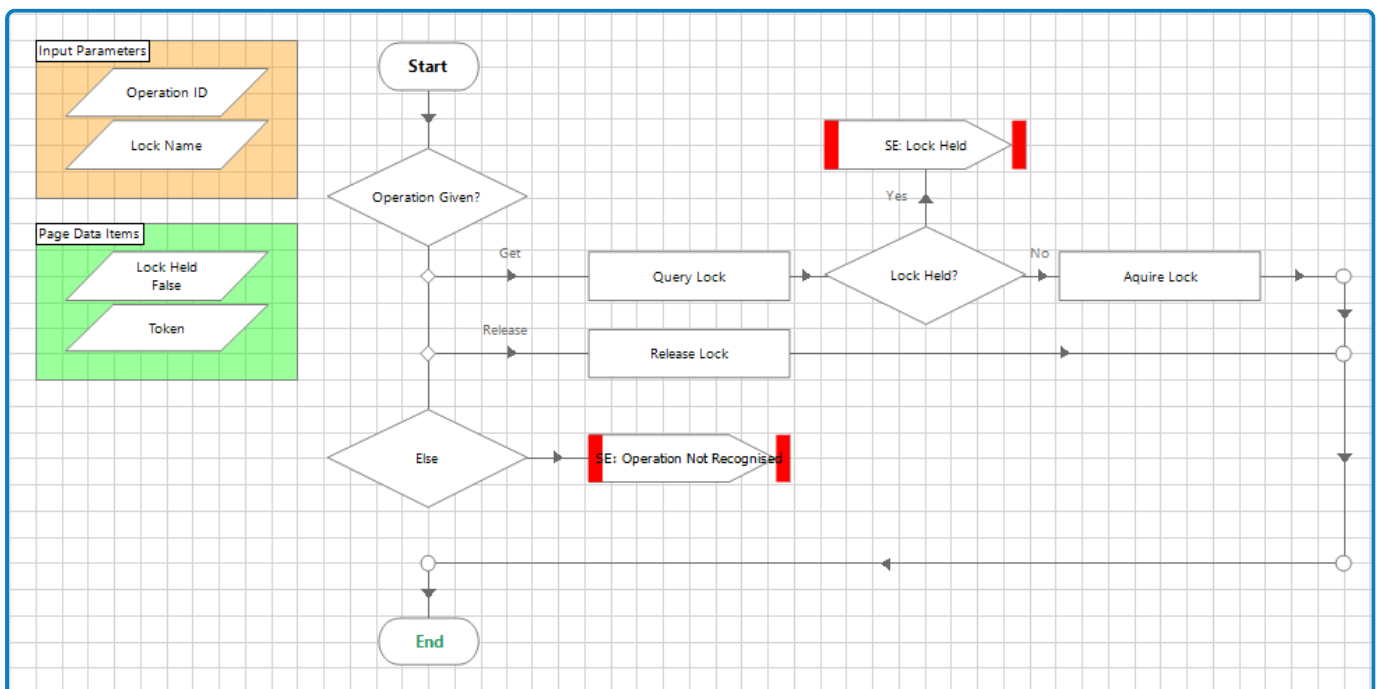


Configuration required:

Update the **Add to Queue** action inputs to match your requirements.

Collector template: Locks

The Locks page handles the environment lock functionality. This enables the process to call different lock actions.

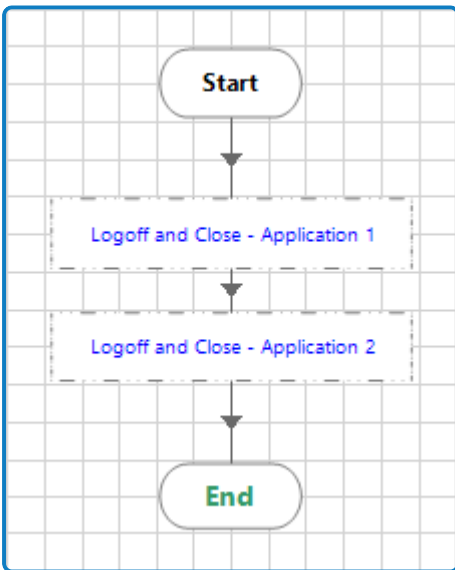


The following choices, driven by the **Operation ID** input parameter, are defined for environment locks:

Operation ID	Description
1: Get	Queries for the specified environment lock using the Lock Name input parameter. If it is not held, it is acquired. If it is held, an exception is triggered.
2: Release	The Release action is used to release the given environment lock.

Collector template: Close Applications

You must create a Close Application page for each application opened when the automation runs. These pages then need to be called from the Shutdown Application Control page. This page ensures that each application is closed when moving on to the next application, or when a request stop is done.



For applications where it is necessary to ensure that a user account is logged out before closing an application, you must use a safe close rather than a straight termination of an application.


Configuration required:

Add the necessary stages or pages to log off and close all applications that the Collector process has interacted with.

Executor template

The *Process - Director - Executor Template* contains the set of business automation steps to be performed on the work queue items populated by the Collector process. This process:

- Launches the application, if required.
- Retrieves the data from the work queue for the item that has been advised by the Director service in the Orchestrator process.
- Performs the business automation steps.
- Marks the item completed or as an exception based on the defined business process logic.

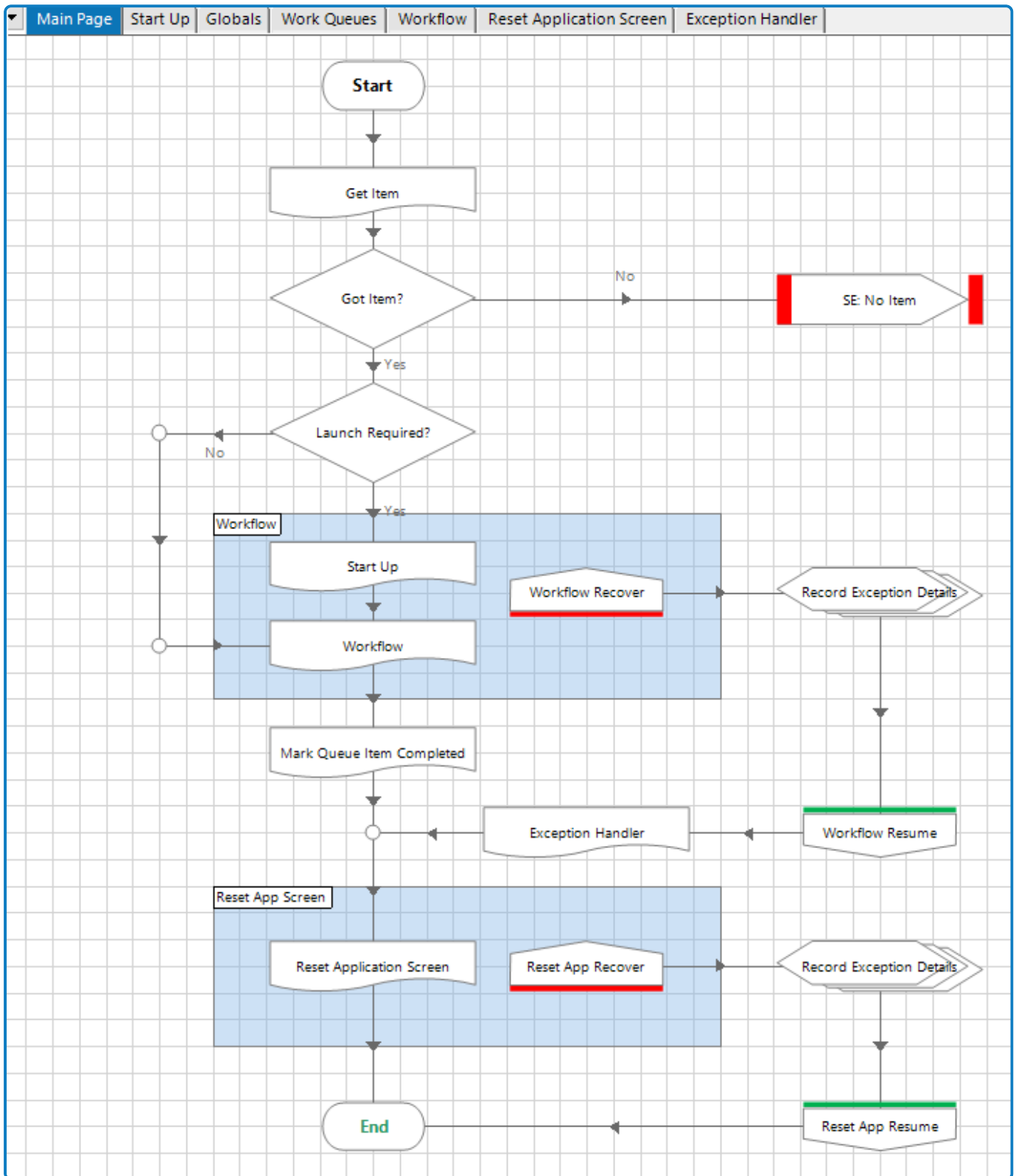
 Any existing executor automations that you intend to use with Director, that have not already been converted when migrating from IADA to Director, will need to be amended to include the required configuration, as described in this section.

This section describes the Executor template, including details of which pages and settings need to be configured for your environment; and which pages and settings can remain static. The following pages are included in the process:

- [Main Page](#) (required) – The foundation of the Executor process, referencing the key stages required to execute the end-to-end process.
- [Start Up](#) – Launches and logs into the application, and then performs the navigation steps required to reach the work screens.
- [Globals](#) (required) – The central location for global variables for this template.
- [Work Queues](#) (required) – Provides the functionality required to manage work queue data. The Operation ID is passed as an input to this page, and based on the Operation ID different actions are performed:
 - **Operation ID 1** – The Item ID, derived from Advise Next Best Item, is passed from the Orchestration process to the Executor process. The Executor process gets the item data by using the Work Queues Business Object action called **Get Item by ID**. The **Get Item by ID** action is performed and item data is retrieved and is populated on the Global page.
 - **Operation ID 2** – The item is marked as completed.
 - **Operation ID 3** – The item is marked as a system exception.
 - **Operation ID 4** – The item is marked as a business exception.
- [Workflow](#) (required) – Contains business object actions and/or process logic steps that are required to perform the automation steps on the item, in sequence.
- [Reset Application Screen](#) (required) – Ensures that the target applications are on the correct screen ready for processing the next case.
- [Exception Handler](#) (required) – Marks the item as either a system or business exception and takes any steps required.

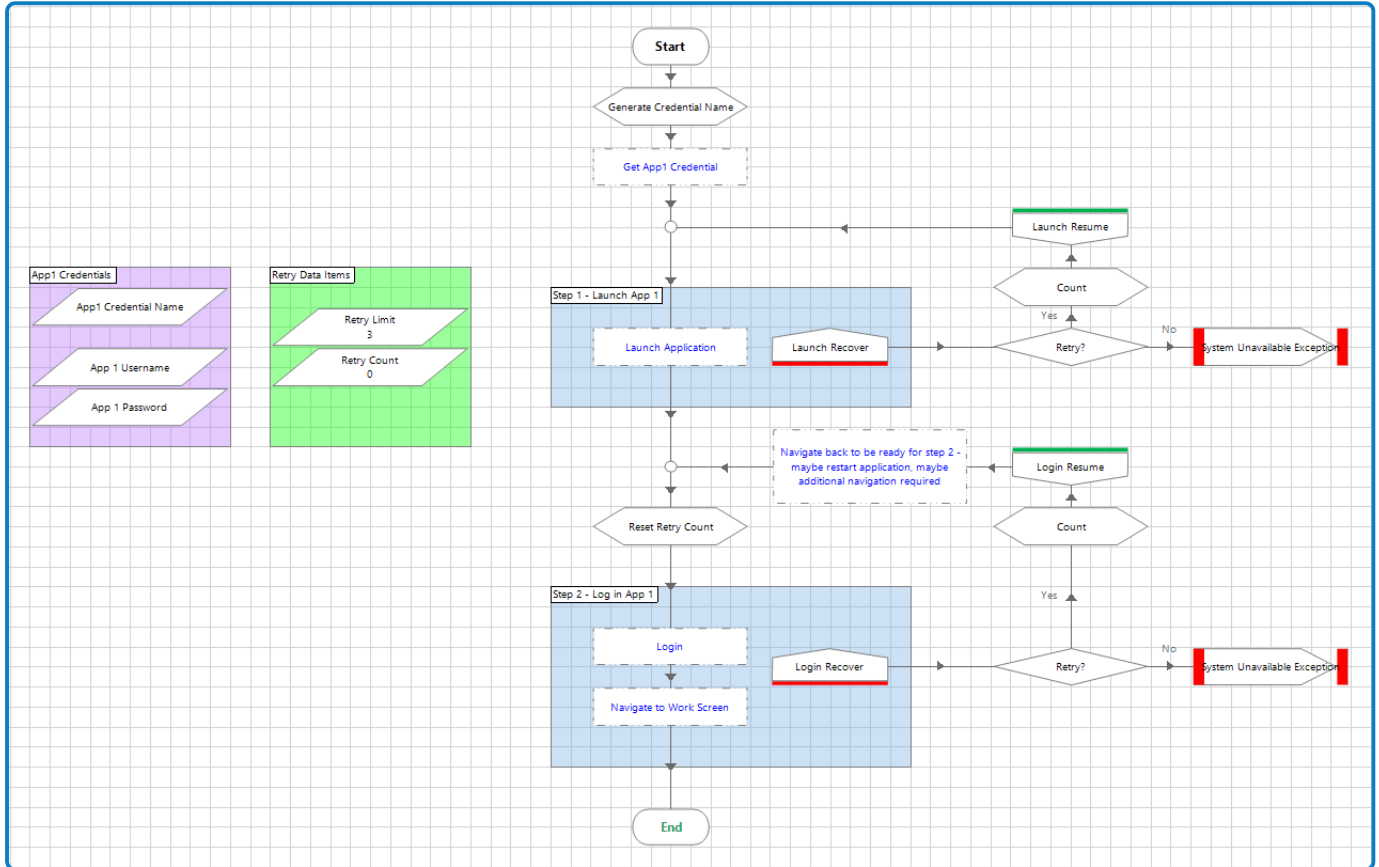
Executor template: Main Page

The Main Page is the foundation of the Executor process and controls exceptions, and the order in which the pages in the template are processed. Each page in this process is detailed below.



Executor template: Start Up

The Start Up page is used to launch the applications required for the Executor process before the process attempts to interact with those applications. A **Launch Required** flag is passed from the Orchestrator process. If the flag is *true* this page is processed; if the flag is *false*, this page is skipped.



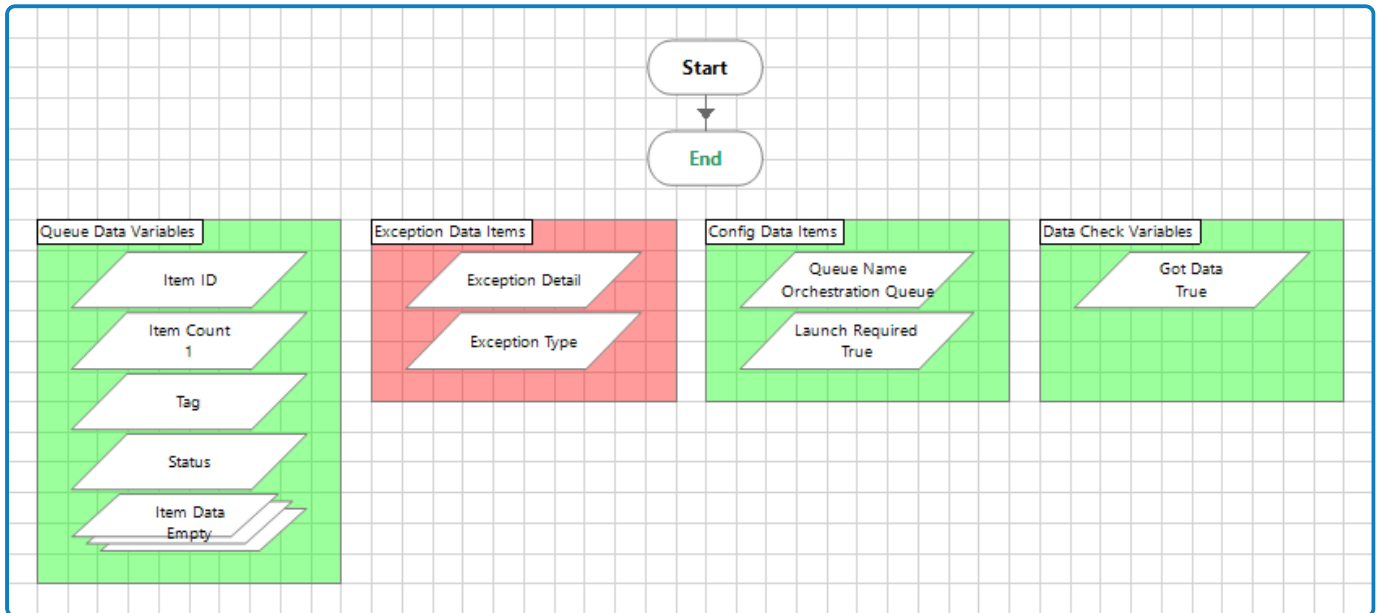
Configuration required:

Add the necessary stages or pages to launch and log into all applications that the Executor process will interact with.

Executor template: Globals

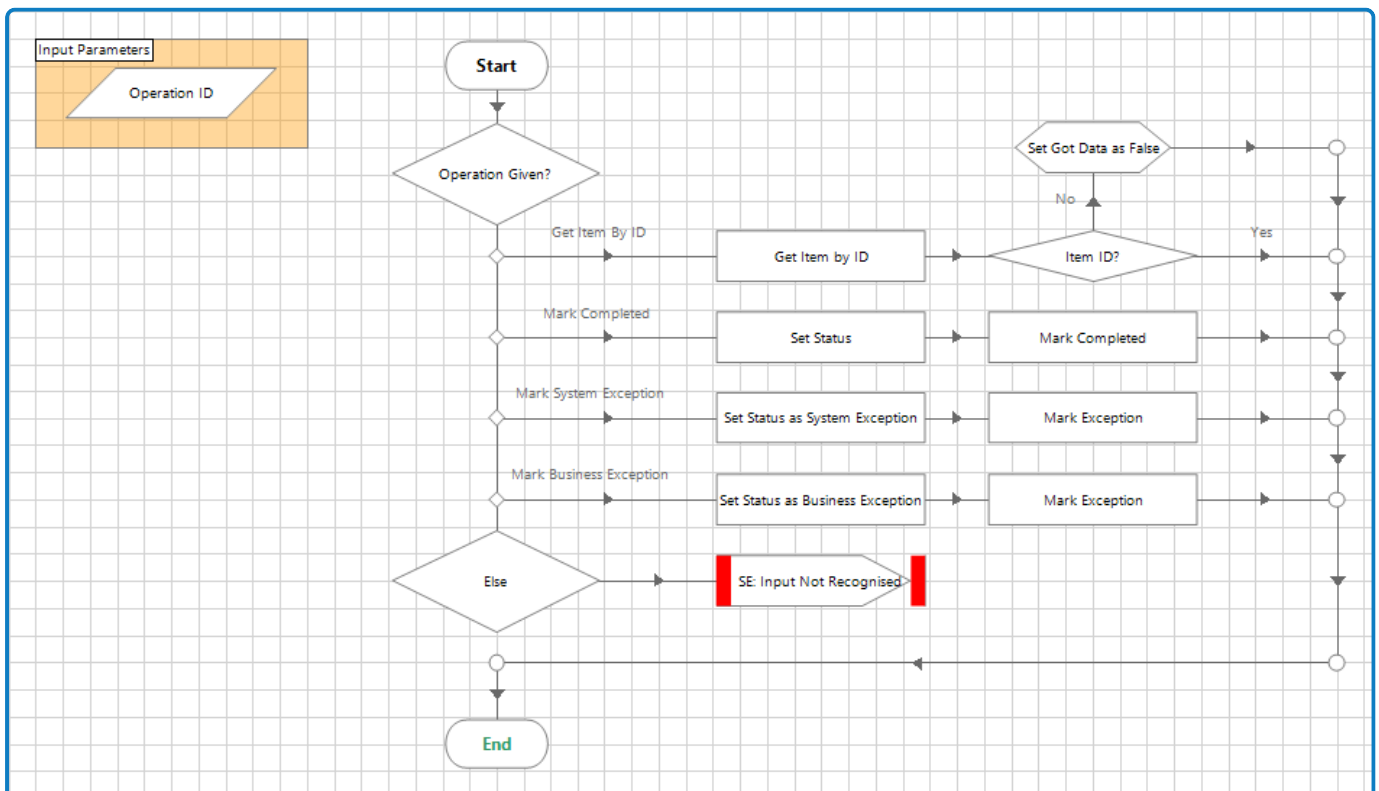
The Globals page provides a distinct, central location, for the storage of global variables.

The **Launch Required** flag is an input parameter of the Executor process which is passed from the Orchestrator process. It has an initial value of *true*. A value of *false* is passed if the Orchestrator process establishes that the required application has already been launched.



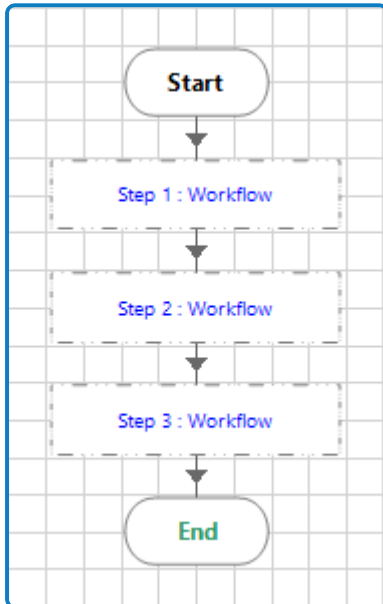
Executor template: Work Queues

The Work Queues page is used for all operations related to the work queue. The **Get Next Item** action is used to get Item details from the work queue by providing the Item ID, which is received from the Orchestrator process. Depending on whether the item is successfully processed, the item is marked as completed or as an exception.



Executor template: Workflow

The Workflow page is where the automation work is actually performed and will need to be configured to perform the specific business process. Each workflow step is processed in a sequence, with each stage performing a specific part of the overall automated process.



Configuration required:

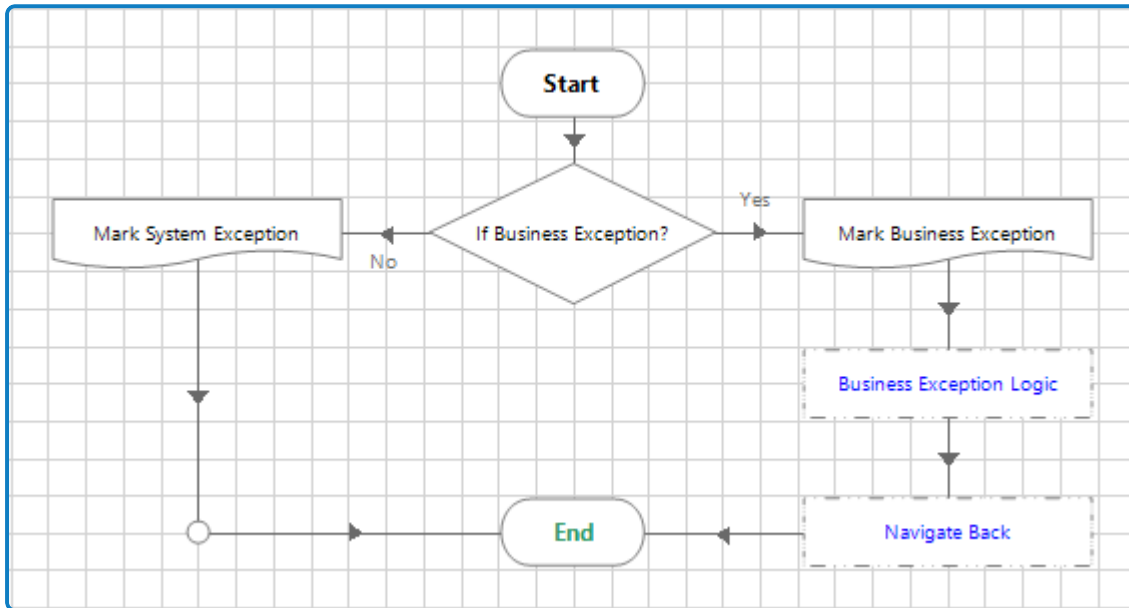
Add business object actions and any other logic to perform the automation steps on the work queue item.

Executor template: Reset Application Screen

The Reset Application Screen page confirms whether the target applications are on the correct screen ready for processing the next item. For example, each time an item has completed processing, the application can be navigated to the home screen to be ready for the next item.

Executor template: Exception Handler

The Exception Handler page is used to handle all exceptions that can be encountered during the execution of the Executor process.



The following types of exception can be encountered:

- **System exception** – This is an unexpected error that occurs during the execution of the process. It could be an application issue that is preventing an application interaction, which will cause a timeout when attempting to identify an element defined in the application model.
- **Business exception** – This is a defined piece of logic set for a specific scenario. The key differentiation from a system exception is that a business exception is a planned exception. A system exception is unplanned by nature.

If the current exception is neither of these, the *Else* condition is triggered to send the exception to the Orchestrator process.