

Process Dispatch Framework

USER GUIDE

**Version: 1.1  
Document Revision: 1.1**



Contents

[1. Introduction 3](#_Toc19853565)

[2. Solution Overview 4](#_Toc19853566)

[3. Pre-Requisites and Environment Configuration 6](#_Toc19853567)

[3.1. Additional VBOs 6](#_Toc19853568)

[3.2. Credentials 6](#_Toc19853569)

[3.3. Steps to Implement 9](#_Toc19853570)

[4. Utility – Process Dispatcher 11](#_Toc19853571)

[4.1. Run Process 11](#_Toc19853572)

[4.2. Get Process Status 13](#_Toc19853573)

[5. Customizing the Process Callback Template 14](#_Toc19853574)

[6. Support 16](#_Toc19853575)

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 authorised 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 – 2019**®Blue Prism is a registered trademark of Blue Prism Limited

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, Centrix House, Crow Lane East, Newton-le-Willows, WA12 9UY, United Kingdom  
Registered in England: Reg. No. 4260035. Tel: +44 870 879 3000. Web: [www.blueprism.com](file:///C:\Users\adutton\Documents\Rebranding\Templates\www.blueprism.com)

# Introduction

This framework is a collection of processes, VBOs, work queues, and environment variables that can be used to provide a means for launching Blue Prism processes asynchronously via SOAP request and receiving process results either through polling or a callback.

**NOTE:** This framework relies on the use of a resource pool to support the execution of multiple processes in parallel. It is recommended that you create a dedicated resource pool with enough runtime resources to handle your projected volume of concurrent processes.

For testing purposes, you may configure the resource pool with a single resource PC.

# Solution Overview

This framework includes the following components:

* Processes
  + Dispatched Process Monitor
    - Used to periodically check processes that were started via the *Process Dispatcher* VBO to determine if they have completed.
* VBOs
  + Utility – Process Dispatcher
    - Used to invoke a Blue Prism process asynchronously.
  + Template – Process Callback
    - A template VBO that should be customized based on your specific callback needs.
* Work Queues
  + Dispatched Processes
    - A work queue used by the *Process Dispatcher* VBO and *Dispatched Process Monitor* process to keep track of processes that have been launched via the framework.
* Environment Variables
  + Process Dispatcher Credential Name
    - A variable that holds the name of a credential, defined in Credential Manager, that will be used by the *Process Dispatcher* VBO to invoke the specific Blue Prism processes.
  + Process Dispatcher Resource Pool
    - The name of the resource pool the framework will use when executing processes.
  + Queue: Dispatched Processes
    - A variable that holds the name of the work queue that keeps track of processes launched via the framework.

The following graphics provide a high-level conceptual view of how the framework functions.



Figure - Dispatching a Process



Figure - Process Monitor and Callback

# Pre-Requisites and Environment Configuration

## Additional VBOs

This framework leverages the following additional Blue Prism VBOs which are available on the Blue Prism Digital Exchange:

* Utility – Blue Prism Process Info
* Utility – JSON
* Data – SQL Server

Ensure these VBOs are deployed in your Blue Prism environment, otherwise you will encounter exceptions.

## Credentials

This framework utilizes credentials in three areas:

* Web Service Invocation
* AutomateC Execution
* Querying the Blue Prism Database

### SOAP Web Service Invocation

The following VBOs must be exposed as SOAP web services within Blue Prism:

* Utility – Process Dispatcher
* Utility – Blue Prism Process Info

Invoking Blue Prism processes or VBOs requires that credentials be provided by the caller for HTTP Basic Authentication.

To set up a user account for use with the web services:

1. Start the Blue Prism Interactive Client
2. Go to ***System → Security → Users***
3. Right-click and select ***Create User***
4. Fill in the various attributes for the user account.
5. From the list of Blue Prism roles, select ***Web Service Consumers***. This role provides the minimum necessary permission to be able to execute Blue Prism processes or VBOs exposed as SOAP web services.

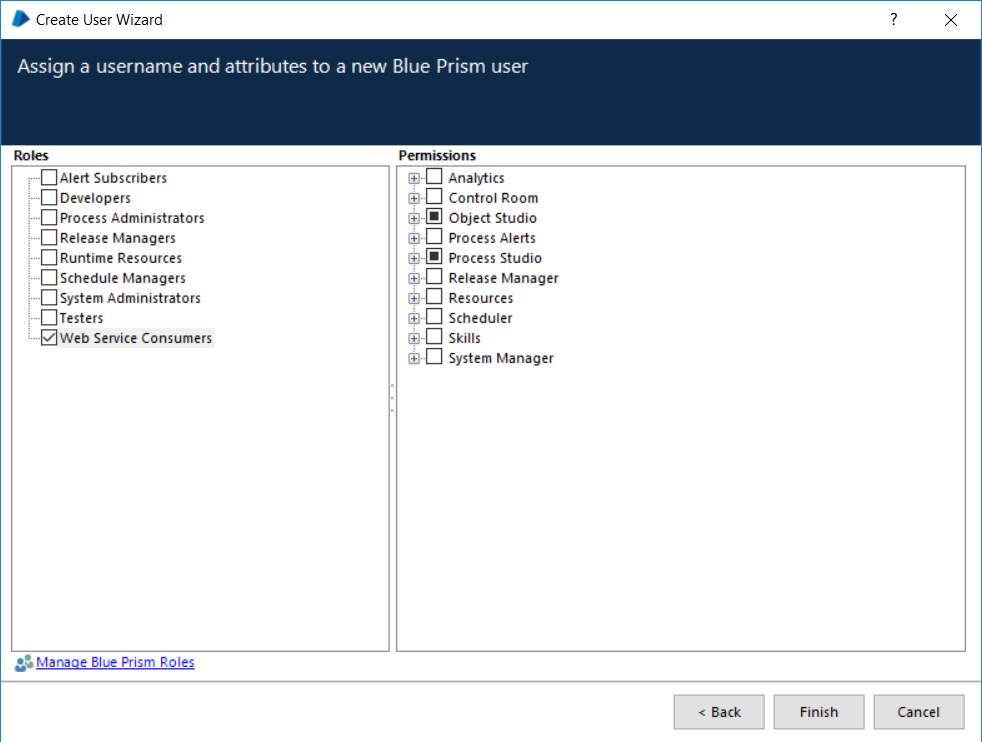


Figure - User Role

If your Blue Prism environment does not include a Web Service Consumers role, you can create it with the following steps:

1. Go to ***System → Security → User Roles***.
2. Click the ***Create*** button.

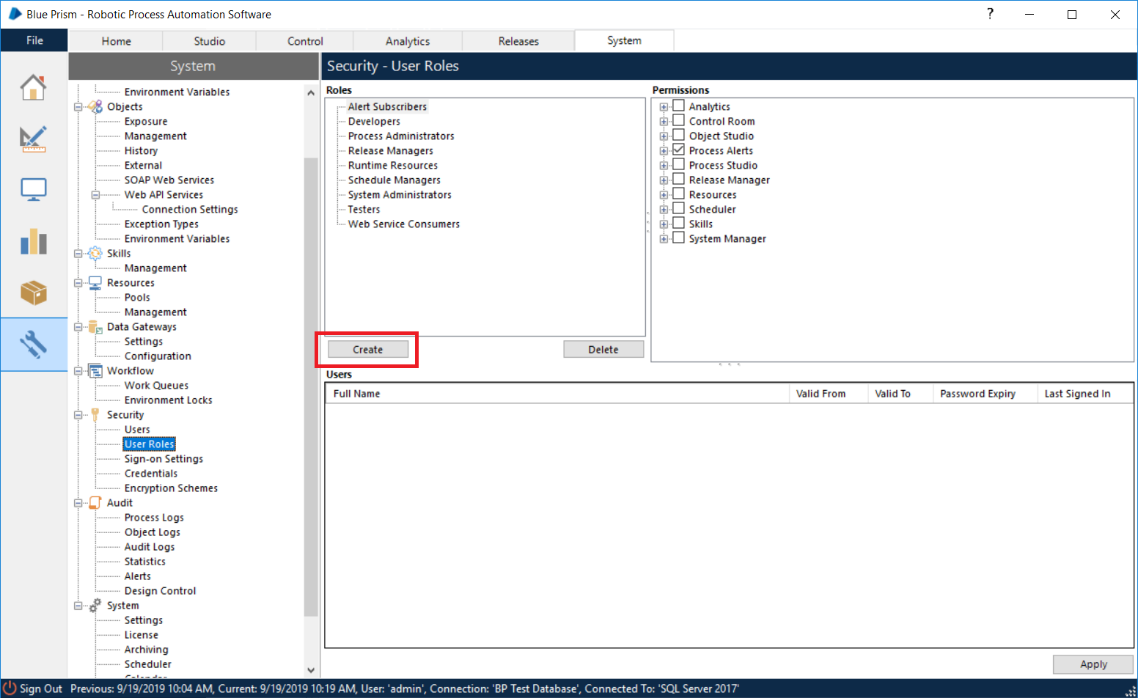


Figure - Creating a Role

1. Provide a name for your new role (ex. Web Service Consumers).
2. In the right-hand pane, expand the tree for ***Object Studio***.
3. Place a check in the box next to ***Execute Business Object as Web Service***.
4. Expand the tree for ***Process Studio***.
5. Check the box next to ***Execute Process as Web Service***.

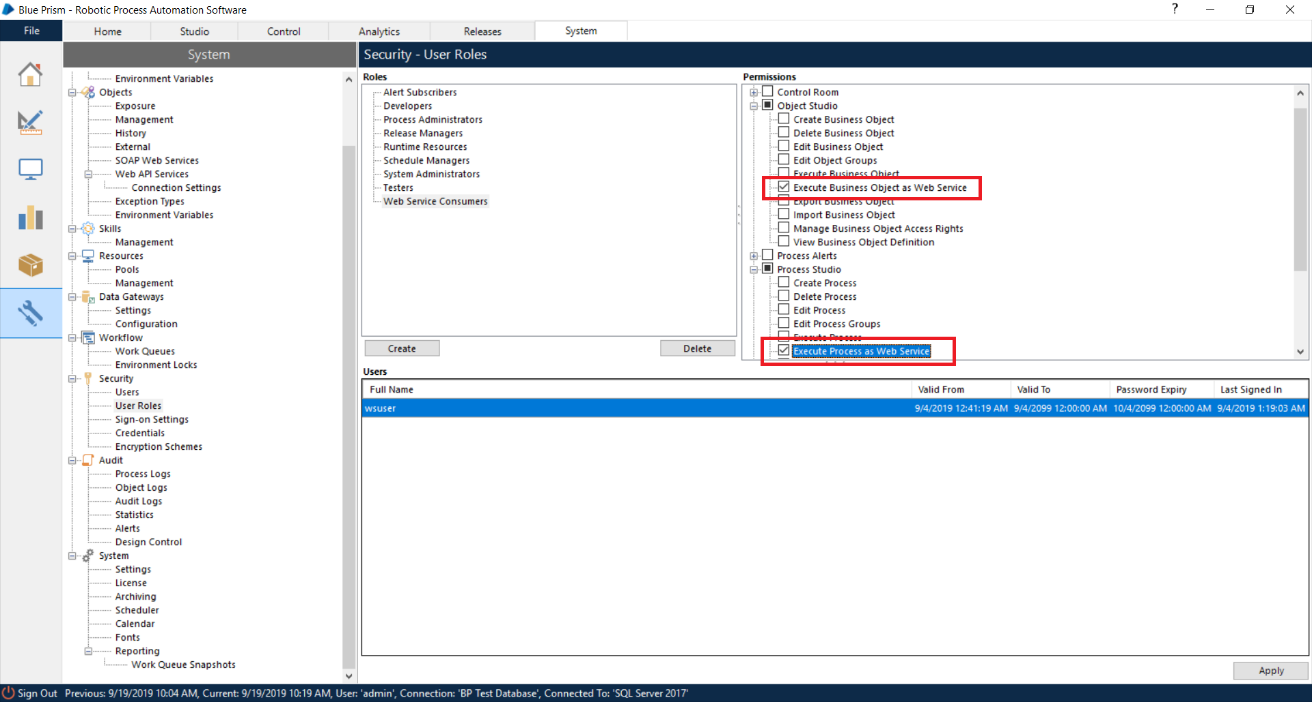


Figure - Selecting Available Permissions

1. In the bottom right corner of the UI, click ***Apply***.

Congratulations! You now have a role available specifically for web service user accounts. Return to section 3.2.1 and complete the steps for assigning this role to your dedicated web service user account.

### AutomateC Execution

AutomateC is a command-line utility included with Blue Prism. It provides various capabilities, but to use the utility you must provide a user account with sufficient permissions. The Process Dispatch Framework uses AutomateC to launch the requested processes and query the status of a dispatched process.

Follow the steps outlined in section 3.2.1 above to create a user account for use with AutomateC. For this user account, set the users role to ***Runtime Resources***.

### Querying the Blue Prism Database

This framework requires a user credential defined on the Blue Prism database. This credential requires READ-ONLY access to the database. If you are unfamiliar with creating user accounts within SQL Server, check with your local SQL Server database administrator.

**NOTE:** The framework does not write, delete, update, or otherwise change any information within the Blue Prism database.

### Credential Manager

The details of the various credentials, described above, must be stored in the Blue Prism Credential Manager. Use of Credential Manager allows the processes and VBOs of the dispatch framework to access the username and password in a secure manner without having to hardcode those values within the actual processes/VBOs.

To add those credentials to Credential Manager, simply create a new ***General*** credential for each of the user accounts and populate the Username and Password with the account details. Within the various processes/VBOs of the framework simply reference the credential name where required. The processes/VBOs will then request access to the credential from Credential Manager.

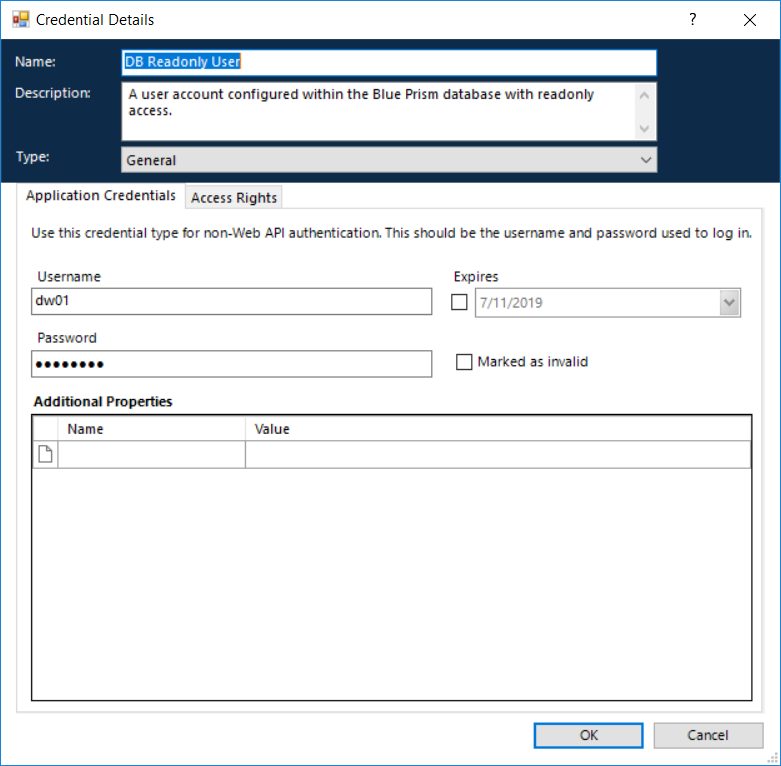


Figure - Credential Manager

**NOTE:** You must set the ***Access Rights*** of the credentials such that they can be accessed by the processes/VBOs of the framework.

## Steps to Implement

To deploy the Process Dispatch Framework:

1. Import the additional VBOs described in section 3.1.
2. Import the Process Dispatch Framework .bprelease.
3. Create the user credentials described in section 3.2.
4. Create a resource pool with sufficient resource PC’s assigned to handle your maximum projected concurrent process volume.

**NOTE:** For directions regarding how to create a Resource Pool in Blue Prism, refer to the [*Resource Pools and Groups*](https://portal.blueprism.com/documents/standard?title=resource+pools) guide on the Blue Prism Portal.

1. Expose the utility VBOs: ***Process Dispatcher*** and ***Process Info*** as SOAP web services.

**NOTE:** For details about exposing a Blue Prism VBO as a SOAP web service refer to the *Web Services User Guide* available on the Blue Prism Portal.

1. Open the ***Dispatched Process Monitor*** process. On the ***Get Process Results*** tab, set the values of:
   1. **DB Server** – The name of the SQL Server instance (ex. localhost\SQLEXPRESS)
   2. **DB Name** – The name of the Blue Prism database within the SQL Server instance.
   3. **DB Credential Name** – The name of the credential you created in Credential Manager to hold the read-only database user account information.
2. Open the VBO ***Template – Process Callback***. On the tab, ***Execute Callback***, implement the necessary logic to support performing a callback in your environment.

**NOTE:** The implementation of the ***Execute Callback*** method is entirely dependent on your environment and what applications you will be using to invoke Blue Prism processes. There is an example implementation based on IBM’s Business Automation Workflow environment. This is simply an example.

1. Verify the following environment variables have values:
   1. Process Dispatcher Credential Name
   2. Process Dispatcher Resource Pool
   3. Queue: Dispatched Processes
2. Define an execution schedule for the process ***Dispatched Process Monitor*** (ex. every 5 minutes).

# Utility – Process Dispatcher

Below are the available actions in the Process Dispatcher VBO and the details of the inputs/outputs that are available.

## Run Process

Use this action to execute a specified Blue Prism process.

**NOTE:** The specified Blue Prism process must be published in the Blue Prism Control Room.

### Inputs

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| UseSSO | Flag | A flag indicating whether to use single sign-on or not. If SSO is not used, a credential name must be supplied. |
| BPCredentialName | Text | The name of a credential, defined in Credential Manager, with sufficient privilege to execute a Blue Prism process. |
| ProcessName | Text | **Required** – The name of the Blue Prism process you want to execute. |
| ProcessParameters | Text | An XML-encoded string containing any input parameters required by the specified process. The XML should consist of a top-level, or parent, *<inputs>…</inputs>* element that contains a number of *<input />* child elements, one for each input parameter of the specific process. Each *<input />* element must include three attributes: ***name***, ***type***, and ***value***  where ***name*** is the name of the parameter, ***type*** is the Blue Prism-specific type of the parameter, and ***value*** is the actual value of the parameter.  *Ex.  <inputs><input name=’Loop Count’ type=’Number’ value=’5’ /></inputs>*  If the process requires a Blue Prism Collection as input the XML of that parameter will change slightly. Instead of a single ***value*** attribute, on the *<input/>* element, you will have a number of *<row>* child elements within the *<input>…</input>* element. Furthermore, each <row>…</row> element will contain a number of *<field/>* children equal to the number of fields within each row of the collection.  *Ex.*  *<inputs>  <input name=’Data’ type=’Collection’>  <row>  <field name=’Name’ type=’Text’ value=’John Doe’ />  <field name=’Address’ type=’Text’ value=’111 Somewhere Dr.’ />  <field name=’Age’ type=’Number’ value=’30’ />  </row>  <row>  <field name=’Name’ type=’Text’ value=’Sally Smithers’ />  <field name=’Address’ type=’Text’ value=’2 Under The Sea St.’ />  <field name=’Age’ type=’Number’ value=’25’ />  </row>  </input> </inputs>* |
| CallbackInfo | Text | A string containing any information you wish to pass through to the callback VBO when the specified process completes. This data will be passed straight through the callback function. No processing or validation will occur within the Dispatcher. |

**NOTE:** You may pass JSON encoded data via the ***ProcessParameters*** variable, but you must pay close attention to the format and naming. Notice how each input variable name starts with an “@”. This is required as the JSON will be converted to XML prior to being passed to AutomateC. The “@” indicates that when the JSON is converted, the item name should be created as an attribute on the specific XML element.

Ex.

{  
 'inputs': {  
 'input': {  
 '@name':'Loop Count',  
 '@type':'Number',  
 '@value':'5'  
 }  
 }  
}

### Outputs

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| Success | Flag | A flag indicating whether the action completed successfully. |
| Session ID | Text | The unique identifier of the process session instance. |
| Output | Text | The raw output of the AutomateC invocation. This may contain error information in the event of an error. Otherwise, it will contain the session ID and information about which runtime resource the process is executing on. |

## Get Process Status

This action will return the status of a specific session.

### Inputs

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| UseSSO | Flag | A flag indicating whether, or not, to use single sign-on. If SSO is not used, you must provide a credential name. |
| Credential Name | Text | The name of a credential defined in Credential Manager. |
| SessionID | Text | A unique session ID for a specific process instance. |

### Outputs

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| Success | Flag | A flag indicating whether the action completed successfully. |
| Process Status | Text | One of the following status messages:   * Pending * Running * Complete |
| Output | Text | The raw output of the AutomateC invocation. This may contain error information in the event of an error. |

# Customizing the Process Callback Template

Once a process, that was launched via the Process Dispatch Framework, has completed, the Dispatched Process Monitor will invoke the ***Execute Callback*** action on the Process Callback Template. You must customize this action to implement the necessary logic to notify your initiating platform (ex. IBM Business Automation Workflow, Oracle Integration Cloud, RedHat Process Automation Manager, etc) that the Blue Prism process has completed and to pass along any available output values from the process.

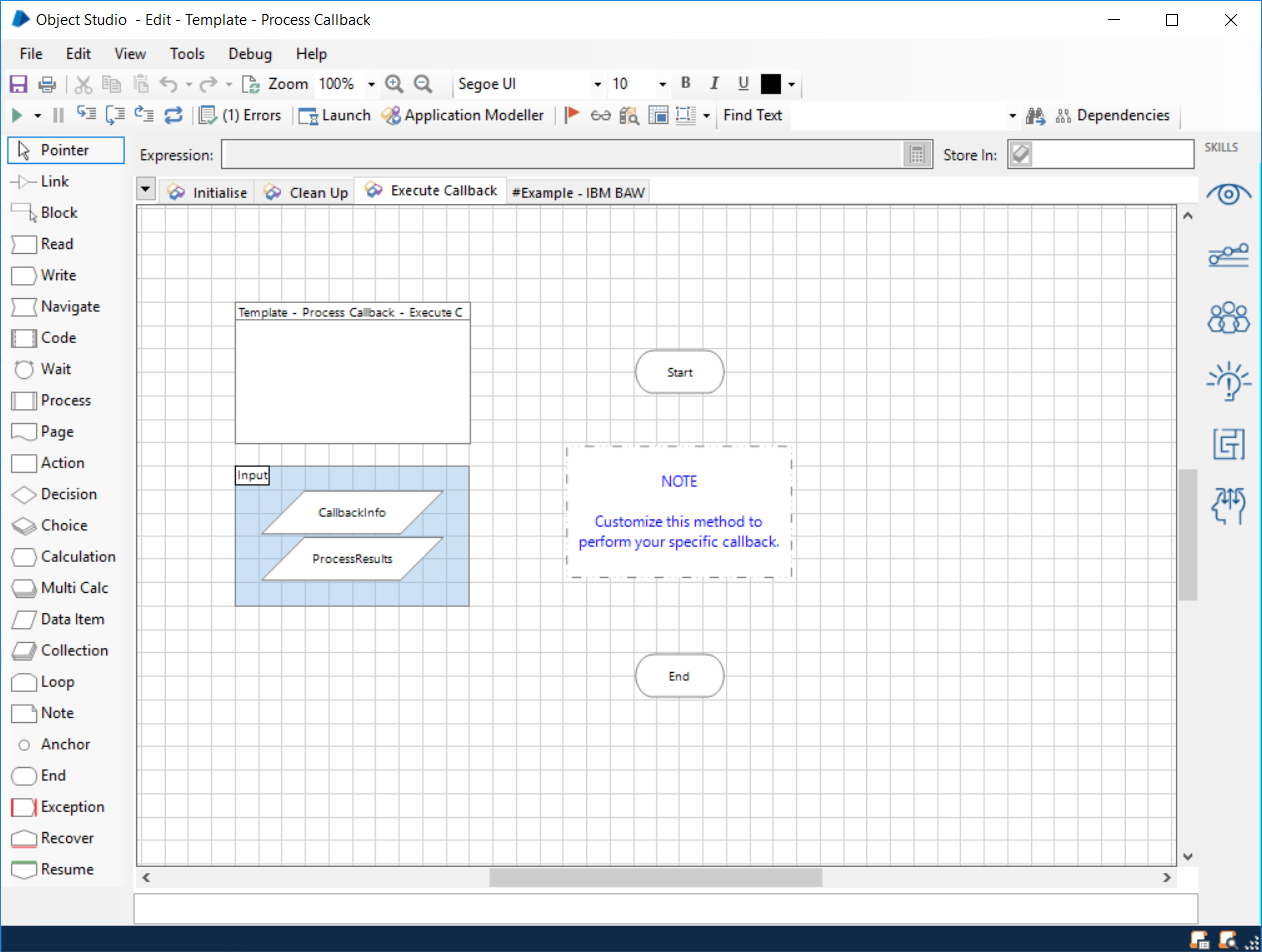


Figure 7 - Execute Callback Action

Within the Process Callback template VBO, there is an example callback implementation for IBM Business Automation Workflow. This is an example of a callback process as it relates to IBM DBA. You will need to implement the logic necessary for your initiating platform.

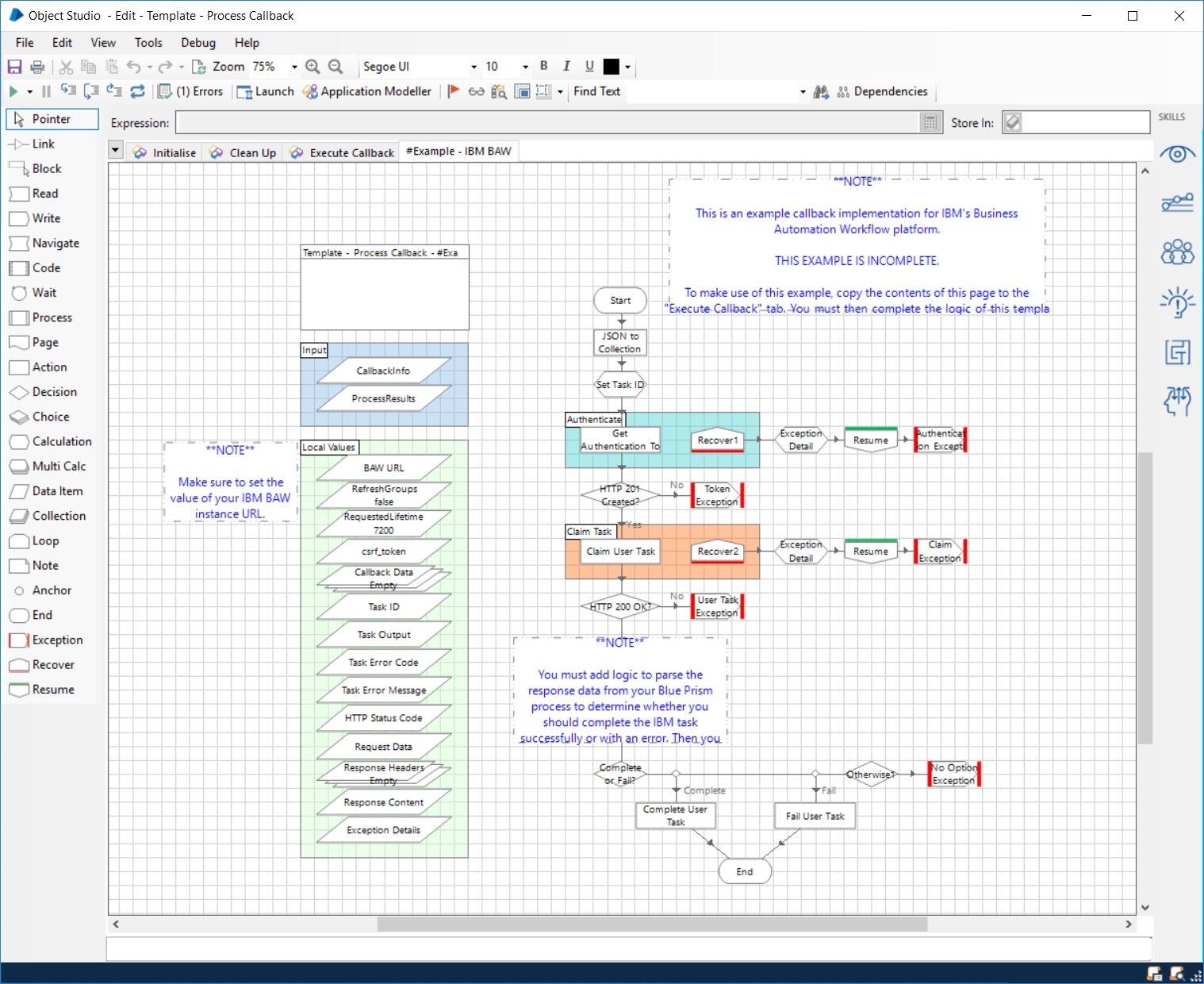


Figure 8 - IBM Digital Business Automation Example Callback

# Support

This VBO is an open source project available on GitHub. For any issues please open a ticket via the “Issues” page on the GitHub repository:

<https://github.com/blue-prism/ProcessInfo>

Additionally, support can be requested via the Blue Prism Community Portal at the link below:

<https://blueprism.connectedcommunity.org/home>