WITHYOU
WITHME

RPA Blue Prism Developer Course

Written Guide

Contents

Intent

This document is designed to augment your online learning experience and provide you with the opportunity to review and revise the content that has been discussed during the individual lessons of the course

Feedback

Please contact your WithYouWithMe RPA instructor if you have feedback on this document or any of your WYWM courseware.

Incorporating Work Queues

Overview

Hi and welcome to the lesson where we will discuss the incorporation of work queues

In this lesson we'll introduce the new concept of work queues and how they are used in the control room. This will give us the knowledge and skill to run the processes we've built as is done out in industry. In this lesson it's also important to reference the Blue Prism Work queues training guide which can be found in the materials tab of this lesson.

As usual we'll switch between the slides and Blue Prism. To get the most out of the video, make sure you follow along in Blue Prism and pause the lesson if you need to.

Work Queue Introduction

Almost every Blue Prism Process will make use of a work queue because as we've mentioned previously, processes and objects are only run in the studio during the building, testing and troubleshooting phases. During business as usual or BAU operations, this is all done in the control room, using work queues.

If you ever need to cycle through data in Blue Prism, you need to use a work queue IAW BP's best practice (for example, you should never just cycle through a collection in your process to achieve a task in the production environment)
This is because uploading your data to a work queue allows BP to track the task which is essential for auditing purposes

Work queues are essentially a list of jobs or cases that are visible in the control room. New cases can be fed into the queue and the queue can be updated with results as each case is processed.

A work queue is an internal and configurable list that enables a process to manage its workload. A process can use different work queues and work queues can be shared by multiple processes just as was the case when we were first introduced to business objects.

Work queues also provide the functionality to store, manage, share and report on process work. A work queue provides the ability for multiple machines (or what we'll refer to as resources), to work from the same queue at the same time, each retrieving cases to work on. For example, "the computer you are sitting on right now has BP installed on it, so we can consider it a "resource".

If there are too many cases for one instance of Blue Prism to cope with or there are thousands of lines of data, we can run the same process on multiple machines to share the load. In this case multiple machines all running the same process will pick up cases from the same work queue while ensuring that the same case (or line of data) is not processed twice.

In the real world we expect that we will be processing lots of data, and because of this we will come across some problems and accept that while the majority of cases might succeed, there will always be some errors. In any case, we will need to record the results, and this is done in the work queue by way of marking an item as complete, or marked it as exception if it could not be completed.

As we've touched on before, queued work can be monitored and maintained from the control room that is available as a tab on the main screen of Blue Prism. The control room is the area in Blue Prism where processes that have been designed, tested and deployed are controlled in a real world environment.

The environment within the control room displays processes, resources and sessions. We know what a process is, and we just mentioned that a resource is a computer or machine but a session represents a process which is currently assigned to a resource and it will maintain a state of pending, running or completed.

Workload and performance management can be based on details that include volumes, performance levels and exception details, and this data can be extracted from the work queue.

In the control room, the queue management tab provided the user an interface for work queues. As you can see, the tab is divided in two. A list of queues is shown in the upper half of the tab and queue contents or a list of items in the queue is visible in the bottom half.

Using a Work Queue

Now let's jump over to Blue Prism and start using work queues in a practical environment. We'll start by having a look at the control room. We can see Queue Management and if we select Queue 1 we can see that our queue is empty.

Now we'll open a process I've created earlier, called Load Queue Process. You can see I have some data in my collection. I have 4 fields, Account ID, Item Ordered, Date and Quantity, and as you can see in initial values, we have 6 rows of data.

Now, we want to add this data to a queue so several resources can be working on it simultaneously. In a process, you can interact or manipulate work queues using pre-built Business Objects, and how do you call on these Objects? We use Action stages.

So first we'll need an action stage so we drag that across, make it a bit bigger and we'll set it up.

We'll call it add to Queue and we'll select what we call an IBO or Internal Business object an IBO is the same as a visual business object except for that fact that we can't amend an IBO and it is only used for interfacing with Blue Prism Functionality, in this case, adding data to a queue.

Now we tell our action stage where we want to send our data, in this case “Queue 1” noting the use of inverted commas because this in an expression window as denotes by the calculator symbol., and we want our data to come from our Order Data collection so we drag that across. Now we click ok, we’ll link it up, click reset and run our process. Now that our process has run we’re expecting that the data from our collection has been sent to our work queue, so let’s go over to the control room and check.

We make sure Queue 1 is selected, click the refresh button in the top right corner of the page and we can see that there are now 6 items in the queue for queue 1, so we’re happy that worked and now we know how to add data to a work queue.

Now we’ve got some data in our Queue 1 work queue, let’s create a new process that will use this data. We’ll call our process Queue exercises. Now that we’re on the main page, let’s add an action stage, We’ll call it get order data and we’ll use another action in our Work Queues IBO called Get next item. We need to again, identify the queue we’d like to pull our data from, in our case Queue 1 in inverted commas. So the queue is our input to this action but we will also have an output which will be the place that we intend to store the data that we pull from the queue. So we click the outputs tab and to extract all the data that’s available, we simply click this button and an associated data item or collection will be created. So we do that for each line item and click ok.

Now we’ll organise our page a little better and make our data items and collection a bit bigger. Long Pause Then we’ll link it up.

So now if we reset and press go, what do you think will happen? Well if we’ve set everything up correctly when our process is run, it will pull the first line for data from our queue and input it into our data items and we’ll see those values on the right of our process page.

Let’s give it a go

Great, we can see that we have some data in our collection. Nothing in our initial values because it started off empty but now we can see the details associated with our first order. We can also see the Item ID which is the unique identifier (also known as a Globally Unique Identifier) that Blue Prism gives every action that it undertakes and this is very handy for traceability and auditing purposes.

Now if we jump across to our control room, make sure queue 1 is selected and press refresh we can see that the first line of data in our queue is locked. It’s locked because we’re still sitting at the end stage in our process and it’s not yet been marked as complete. This lock feature prevents other resources from trying to process this line of data and will therefore prevent the duplication of work and data entries.

You can also see that the other 5 rows are pending as indicated by the three blue dots. In the event that another resource came along right now, it would skip the locked line item and move on to the first line it could see that was pending.

Ok, so let's reset our process back in the process studio, and then if we return back to our control room, highlight Queue 1 and refresh, we can see that the process is now complete, however it's been marked with an exception.

Let's do this again in the process studio for the next row. Again we can see that data has been added, and if we go back into the control room we can see that the second line of data has been completed with the exception type and date annotated on the right side of the table.

Marking Work Queues

Overview

Hi and welcome to the lesson where we will be advancing our knowledge of work queues

In this lesson we'll build on our knowledge of work queues and we'll learn how to process lines of data without exceptions.

This time we'll jump straight into Blue Prism, but to get the most out of the video, make sure you follow along and pause the lesson if you need to.

Marking Work Queue Items

So let's start by opening up our control room. At the end of the last lesson we were able to process two lines of data using our Queue Exercises Process but after we processed each line it was maked with an exception rather than being marked as complete. Now we're pretty pleased that we know how to send data to and pull data from a work queue but if we go to the effort of processing a line of data we'd prefer that is was marked as complete rather than being marked as an exception, especially given that the work we've done so far has indicated that an exception is not along the happy path and something that we'd prefer not to have.

So how do we process data and mark it as complete rather than as an exception?

Well it's as simple as jumping back into our process and adding an additional action stage that marks our item as completed.

So let's open our Queue Exercises Process, delete the last link and add an action stage into our process.

Now, for our actions stage we'll make it a bit bigger, call it mark complete and we'll use our Work Queues IBO and the mark completed action within it. Now we need to use the data item that houses our Item ID so our action stage knows which line item to mark as complete. So let's drag that in and because we don't have any outputs we can click ok and link it back up.

This time when we reset and click go we're expecting our process to get the next item as it did before, but this time once it's extracted the data from the queue it will mark that line as complete. Let's give it a go.

That looked pretty good but we can see if our process marked our line item as complete but heading back to our control room, highlighting queue 1 and looking at what happened. Great, we can see that the item we juist processed was marked as complete which is indicated by the green tick seen here.

Let's now go back into our queue exercises process and work through the remaining data.

What happened here? We got an error that said invalid item ID. This is because our get next item action stage tried to get the next item but there wasn't any new items left so no data was populated in our data items or collection, our mark complete action stage then tried to mark that the line was complete but because no Item ID came across it threw an error.

End of a Work Queue Functionality

So to avoid getting this error every time we get to the end of a queue what could we do?

You guessed it, we could use a decision stage. This allows us to check whether additional rows exist within our queue, and if not we can proceed to our end stage. Of course if there are additional rows we can incorporate a circular path and go through our process again. So let's set that up now.

We'll delete the link after our get next item and before or mark complete action stage and give ourselves a bit more room. Then we'll drag a decision stage across and make it a bit bigger. We've chosen this location because we need to wait and see whether there's going to be anything in our Item ID data item, which we wont know until after our get next item action stage, but it needs to be before our mark completed action stage or we'll get the same error as we did before when we reach the end of the queue.

So we'll call our decision stage more items?

Now the trigger for our error last time was that there was nothing in our Item ID data item, so given we know that, what should we put in our decision stage expression window?

Well, we want our decision stage to ask, is Item ID equal to something? We don't care what's in our Item ID data item, we just care that something's there, and if it is we'll loop around again, but if there's nothing in our Item ID data item, we know we've reached the end of the queue and we want to proceed out of our circular path. So putting our coding hat on for a second, how do we write, Item ID = something?

The best way i've seen is to first drag item ID across and write = nothing, denoted by these inverted commas. So let's validate that and organise our process a bit better. So essentially, if there's something in our item ID data item it will be less than or greater than nothing and we'll get a yes response from our decision stage meaning we'll then continue on around our loop again. But if our Item ID data item isn't less than or greater than nothing (i.e it's equal to nothing) We'll get a no and we'll proceed straight to end.

Great, now that's linked up, let's go back into our load queue process and load our queue again, noting we got to the end last time, and we'll check that it's reloaded by highlighting our Queue 1 and confirming we have 6 new pending line items.

Great, so lets go back into our queue exercises process, reset and press go. Here we can see that our first action stage is getting the data from the queue as before, the decision stage determines whether anything's been bought across and because it has we continue on to mark that line as complete and loop around again.

When we get to the end of the queue and our action can't bring any data across, our decision stage detects this because our Item ID data item is equal to nothing and we then proceed to end. Great.

Now as a final functional check to confirm that all worked, we can go back across to our control room, highlight Queue 1 and we can see that all our items have been processed and marked as completed.

Creating Work Queues

Overview

Hi and welcome to the lesson where we will be taking our knowledge of work queues a little further

In this lesson we'll build on our knowledge of work queues and we'll learn how to create a new work queue from scratch.

This time we'll jump straight into Blue Prism, but to get the most out of the video, make sure you follow along and pause the lesson if you need to.

How to Create a Work Queue

To create a work queue we can go into the system tab, which is the administrative area within Blue Prism. Once here we can find workflow on the left side of the page and underneath that we can see Work Queues. From here we can create a new queue using the new button at the bottom of the screen.

We'll call our new Queue My First Queue, and we'll change our Key name to Account ID which was our unique field in our load queue collection that we used in the previous lessons. It's important to remember here that we need to get this text exact or Blue Prism won't be able to recognise our Key Name field.

We can also choose to encrypt this queue in the event any sensitive data is being processed. Finally we can set the maximum number of attempts and once we're happy with all this we click apply to save our changes.

So to populate this queue let's use our Load Queue Processes. Let's open that up but we'll make some adjustments before our Order data will be sent to the queue we just created. What else will we need to do?

Yep, we'll need to change the name of the queue that we're sending our collection data to. So we'll change that to My First Queue leaving the rest of the properties as they are. Now we click ok, reset and press go.

We can check that our queue's loaded up again (pause) and we then go into our queue exercises process and pull the data from our new queue, but first, what do we need to change?

If you guessed the name of our queue in the get order data action stage you're spot on. We'll change the input to My First queue but the rest is fine so we can click ok, reset, speed it up and run it again.

Now we can see in the control room that our queues been processed and all our items have been marked complete, but we can also see that because we selected account ID as our item key when we build this queue, the item ids associated with each order have also been populated, this gives us a much better idea of which line of data has been processed and very helpful from an auditing, tracking and fault finding perspective.

Work Queue Configuration

Overview

Hi and welcome to the lesson where we will be discussing the configuration work queues

In this lesson we'll build on our knowledge of work queues and we'll learn how to configure work queues so they can be applied to a real world scenario in industry, and maybe also your summative assignment..

Attempts

First let's talk about attempts. Items can be worked more than once if necessary and we therefore have a max attempts field in the work queue configuration section of our system manager, and we saw earlier that we can set the number of times we'd like our queue to work a line of data when we created our work queue. We also saw when we set up our work queue that our attempts are set to one as a default but we can of course set this to a higher value if necessary.

If we set our max attempts to more than one, in the event of an exception our work queue will immediately revive any exception cases by cloning a new item and inserting it into the queue. This will happen as long as there is an exception up to the maximum number of attempts, at which time the queue will continue on with the remaining line items. We can see this on the slide where one of our lines has been attempted twice.

In this case, our first item has been completed, our second line item has had an exception and our max attempts was two so our queue had another go, this produces another exception so we moved on to our third item, which is currently being run and we're yet to attempt the rest of our line items which we know based on the pending icon.

There is however a method for overriding our max attempt selection that we made in the System Manager. We can do this by using the retry input parameter of the Mark exception action within our Work Queues IBO as you can see on your screen. If we set retry to true the queue will generate another attemp as it's done before if the limit of max attempts has not been reached, however if false is set, even if there are more attempts available as per our setting in the work queue our action stage won't allow our queue to carry out any more attempts.

The keep locked input parameter is used in conjunction with the retry parameter and when keep locked is true, the new cloned item in the work queue that was created by our work queue due to an exception, will be locked and therefore only accessible by the resource that ran that line item. This provides us the opportunity to carry on working on the new item that's been created. If keep locked is false our cloned item is freely available to be worked on by another resource that may also be working from this queue.

There is also quite a bit of additional functionality in queue management within our control room. Most of these function are quite intuitive, however we'll briefly highlight some of them here, but make sure to take the time and explore these yourself in your own version of blue prism.

Defer Work Queue

Firstly, the defer work queue functionality allows us to defer new items to prevent them from being worked too soon. If a deferred date is specified when items are created, the queue will hold on to them until the specified date is reached.

This is useful for running queues during non standard hours or potentially when more resources become available. It is possible to change an items deferral date from the control room by right clicking, selecting change deferral date and changing the date as required.

Pause and Resume a Work Queue

We can also pause and resume work queues. A work queue can be temporarily paused so that it is not releasing items at all. The resume functionality as you can imagine resumes a work queue allowing it to reissue line items after it's been paused.

From the perspective of a process a paused queue will appear to have no available items. This feature can be a useful way of making a process think it has no work to do, just as was the case when we'd processed all our line items in the case of our Queue process, it would not detect an item ID and would proceed to the end stage. If however the queue was resumed and we reset and ran our process again, it would start processing data again as we saw previously.

Filtering Work Queues

We can also filter queues. The queue contents list can be filtered to display a subset of the queue in our session management area. We also can do this by manipulating the drop down list that are at the top of our columns. We can also save this filtered view and reapply it as desired.

It's also important to note that a resource can only handle one queue at a time. So if you resource is giving you a warning that it's too busy, this typically means that you have a pending session. So if your session list appears empty, it's a good idea to make sure that you don't have a column filtered that's stopping you from seeing more relevant details.

Reports

Reports is the final thing we'll cover in our work queues tab. In terms of reporting, the queue contents list can be exported as a basic report by right clicking on a queue item in the lower half of the screen and selecting “exporting current view as a report”, and follow the subsequent steps that will help you create a report file.

Make sure you take the time to familiarise yourself with the control room because a developers skill in working with this part of the software grows with exposure.

Startup Parameters and the Control Room

Overview

Hi and welcome to the lesson where we will be discussing Start up parameters and walking through the selection of a resource in the control room

In this short lesson we'll build on our knowledge of the control room to incorporate startup parameters and we'll step through how we assign a resource to a process. This is really applicable to RPA process controller roles out in industry where multiple process are running on multiple machines or resources simultaneously.

As usual we'll switch between the slides and Blue Prism. To get the most out of the video, make sure you follow along in Blue Prism and pause the lesson if you need to.

Startup Parameters

Let's first discuss startup parameters. So far we've been looking at inputs and outputs for actions within an object or processes only. Inputs can also be applied to an entire process itself, enabling a process to accept external values whenever it starts to run.

These process inputs are knowns as startup parameters, so let's jump across to Blue Prism and step through how we set these up and run our process from the control room.

So what we want to do is provide a startup parameter to our circular paths process. So we'll go to the main page of your circular paths process and add an input to the start stage. We'll call it Desired Number of Loops and now we'll set it as a number data type and use the existing max loops data item to store our input.

Now, to run a process with a startup parameter, we need to publish it and run it via the control room. By default, our process is not immediately available in the control room. This is a safety feature of Blue Prism that will stop us from being able to access unfinished processes and running them accidentally.

If we go to the control room tab next to the home screen, you will see that that available processes list is currently empty. To get our circular paths process visible in our control room we need to first publish them. So let's jump back in our circular paths process and on the main page open the properties of our process information stage. You would have likely noticed this details box during your time in the studio to date.

As with everything else, if you double click it you can get to the properties, and you can then check the publish this process to control room check box at the bottom of the window. We then click ok, and we can see this green tick that indicated to us that our process has been published.

We now save our process, close it down and we'll see what we've got back in the control room.

Now we can see that our circular paths process is visible in the available processes section of our control room.

Note that this page is in the session management section and on the right side of the page we have a resources section. This is where we can allocate a resource or resources to a process.

Assigning a Resource to a Process

As we've said, a computer or machine that is running with Blue Prism installed is known as a resource and an instance of a process running on a particular resource is known as a session.

Now in the control room, let's use an available resource, in this case your computer, to run a session, in this case our published circular paths process.

We'll step through it now but while we're going through it, try and also think about the context of a process controller that has multiple processes and multiple resources available at a time.

So we'll start in here in the control room and we'll see that on the right side of the screen we have one available resource which is my computer, for you this will be your computer displayed as a blue icon.

On the left we can see our circular paths process. To assign this process to a resource we drag it across to the blue icon and see that a new session row is added to the environment list at the button of the screen.

You'll note that dragging your blue available resource on to the process which is on the left side of the screen has the same effect.

The session in the button menu is initially displayed in orange to indicate that it's status is pending. I.e. it's ready to start.

We can start the session by selecting the orange row, right clicking and selecting start selection.

You'll notice that because we set an input parameter in our circular paths process earlier, we're prompted to input a number that will be our max loops. When we click start, this number will be sent to our max loops data item and our process will run to the prescribed number of loops based on the number we just inputted.

Let’s enter 100 and press the start button. We can see that the status column changes as the process runs and it eventually becomes blue to indicate that the session is complete. To run this again we can repeat this process again and again.

In reality, processes don’t often use start up parameters because as you’ve just seen, these need to be entered manually from the control room, and this can be labour intensive for the process controller or administrator managing our control room.

We can also stop running a session from the control room if necessary. We do this by right clicking the session we’d like to stop and selecting stop a session.

We can also review session exceptions and session logs. We’ve discussed exceptions in detail thus far, but session logs are essentially a record of each step a process takes and this log is referred to as a session log. We can also access session logs in the control room.