WITHYOU
WITHME

RPA Blue Prism Developer Course

Written Guide

Contents

Contents **1**

Intent **2**

Feedback **2**

Process Layers **3**

Overview 3

Multiple Process Pages 3

Business Objects Overview **7**

Overview 7

Applications and Blue Prism 7

Business Objects 7

Action Stages 9

Building an Object **11**

Overview 11

Microsoft Notepad 11

Application Modeller 11

Navigate Stages 14

Navigating Processes and Objects **16**

Overview 16

Calling on an Object 16

Process Reference Stages 17

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.

Process Layers

Overview

Hi and welcome to the lesson where we will be discussing Process Layers.

As part of our Process layer discussion we'll start to consider some more complex processes and we'll walk through some of the practices we employ within Blue Prism to manage the extra layers of complexity. We'll also talk about layers of logic, organisation of pages and stepping in and out of pages.

As we've done so far, we'll switch between the slides in this video and Blue Prism. To get the most out of the video, make sure you follow my actions in Blue Prism and pause the lesson if you need to.

Multiple Process Pages

So far we've designed and built processes in the process studio using just our main page. This has been great for simple processes but in the real world, we will often see a large amount of flow, structure, stages and data items in our processes. These complex processes require a significant amount of space and if not organised using best practices can quickly become unwieldy.

As is the case with other business flow diagrams, a Blue Prism Process can be spread over several pages that are arranged in a hierarchy. We call the levels of this hierarchy our process layers. We have our main page at the top of our hierarchy, with subordinate layers created below depending on the complexity of our process.

The segregation of discrete bodies of work into individual pages also given us the ability to call on them multiple times without needing to create new process logic each time.

To understand this a bit more clearly let's jump back into Blue Prism and open up our Circular Path process and add a subordinate process layer by adding some pages.

Now we're back in our Circular Paths Process. You'll recall that this process loops around our circular path for a predetermined number of loops annotated in our max loops data item.

So given we're talking about process layers and additional pages, let's add a page. Do you remember from the previous lesson how we do this?

Well, we go up to the tab area and next to our main page tab we right click and add a new page. We'll call this First Page and click ok.

Note that we start off with a start and an end stage, just as we see when we create a whole new process. This page is associated with your main page, but for simplicity you can think of it as a new process that is in a process layer below the process on our main page.

For simplicity, let's quickly set up some data items and a calculation stage to demo the process functionality on this page. (Long Pause)

Now that's done let's return to our main page and from our main page we're going to send our process from the main page to our First Page. Do you remember how we do this?

We grab a page stage that will push our process to another page, and in doing so, step down a process layer, in this case to our First page. Let's set that up now.

We drag our page reference stage onto the main page and when we release it, a new page wizard automatically pops up. This wizard helps us connect this page reference to another page. Because we want to create a reference to our First Page which already exists, we can leave the top row selected, but if we wanted to refer to a page that we hadn't yet created we could do this by selecting the second or third option.

So let's click next. Now we select the page we want this page reference to connect with. Blue Prism has detected that there is only one page that we can connect with and it's the First page that we just created. If we had several pages to choose from they'd also be visible in this window. But we're happy with First Page so let's select that and click finish.

Now this page reference is linked to our First page and when we reach our page reference stage, we will be sent down a process layer into the First page. So let's find a place to put our page reference stage and link it up.

We'll put it after our Loop again decision stage so our process will be sent down a process layer to our First Page each time we cycle through our circular path.

We'll also change our max loops to three just to make the whole exercise a bit quicker for us.

So really quickly, what are we expecting to happen here?

Well press start, we'll come into our count stage, Loop Count will increase by one, it will be less than or equal to 3 so we'll loop again and then we'll be sent down a process layer to our First page. We'll then go through whatever process we've setup in our First Page, so in this case it will be pretty simple and once we get to our end stage we'll be sent back up a process layer to our main page for us to continue on with that process.

Sounds pretty good, let's press reset, check that there's no errors and see what happens.

Great, that worked. You can seen that each time we looped through, we stepped down a process layer onto our First page and each time we go to the end of our First page we jumped back up to our highest process layer on the Main page, continued the loop, and did it again.

Now, we can have several layers of logic depending on the complexity of our process. Let's say we wanted to add a third process layer to this process. How would we do this?

What if we added another page reference to our Main page after our first page reference and called it Second Page? Would that give us a third process layer?

Well no it wouldn't, because both our first and second page are being called on by our main page, meaning that they would both be on the same process layer which is directly subordinate to our Main Page.

We could add a page reference between the start and end of our first page. Would that give us a third process layer?

It sure would, but let's give it a go to confirm this works. We'll go back into our First Page, get another page reference stage and use the wizard to create ourselves another page. We'll call this page Extra Layer and you'll notice that when we create a new page using the wizard, it's automatically connected to the page that we created the page reference in.

Now we'll connect it back up and go across to our Extra Layer page and as we did before, we'll quickly add some data items and calculation stages to our Extra page to demonstrate the process logic and we'll be good to go.

Now we go back to the main page and click reset, but before we press play what do we think is going to happen?

We'll start on the main page which is our first process layer, we'll go through the calculation and decision stages as we did before, then we'll be sent to our first page which is our second process layer. Once we're in the process on our First Page, we'll be sent to our Extra Layer page which as we recall was our third process layer. Once we're down in our third process layer on our extra layer page, we'll do a brief calculation and once we reach end we'll go back up to our second layer which is our First page. From here we go up again to the highest process layer which is our main page. This process will continue for as many loops as we've set.

Now let's speed our process up slightly and press play.
I know we've spoken a lot about pages and layers so before the end of the lesson let's have a quick look at our process represented in a different way that more clearly demonstrates the processes layers within this process.

Here you can see our three process layers and where the three pages within our process sit. Feel free to pause the lesson here and refer back to the process we've built in Blue Prism.

As we said at the beginning of this lesson, when we start to work with increasingly complex processes, separating the logic of our complex business process out on more than one page can help keep our diagrams tidy, structured and readable.

You will also generally not be working alone when developing bots in industry, so segregating discrete bodies of work within a team is often done and it therefore makes sense that these discrete pieces of work are developed on separate pages. This separation also assists with troubleshooting which we'll see later in this course.

Finally, page discipline is very important. We need to ensure that we maintain the path references between our pages, making sure that the correct syntax, case sensitivity and naming conventions are used. Because if we change the name of one of our pages across the top, our page reference stage won't be able to find the page it's referencing unless it's properties are also updated. As we just mentioned, this is where the correct syntax and case sensitivity are really important.

Business Objects Overview

Overview

Hi and welcome to the lesson where we will be discussing Business Objects and the Object Studio.

In this lesson we'll use what we've learnt so far and apply it to the Object Studio to get a better understanding of Business Objects.

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.

Applications and Blue Prism

So far in this course we've demonstrated basic processes that explain the mechanics of Blue Prism functionality but they haven't really done any useful work for us that could be applied to an industry environment.

A real process in industry will be required to do useful tasks and in order to do this, our process will need to interface with applications like Microsoft Excel, Microsoft Outlook, Gmail and a range of other programs.

Business Objects

As we touched on briefly earlier in the course, the interface to an application is not contained within our process diagram, but in a separate diagram that we call a business object. The purpose of a business object is to provide an interface between our process within Blue Prism and an application that we need to use.

Unlike our business processes, a business object is not developed and configured in the process studio where we've been working so far, it's created in a parallel environment under the studio tab called the object studio.

A business object is never used on it's own and is not run through the control room when in the production environment as we saw with our processes. A business object is only ever run by being called upon within a process through the use of an action stage.

Finally, business object logic can be thought of as forming a seperate layer to process logic. Business object pages run one at a time and are usually not connected or sequential like the pages of a business process. Rather, they run in a sequence determined by the order in which they are called by our process.

The object studio is used to capture the functionality of an application so that it can be employed by a process. A process is then used to combine this application functionality with business process rules in order to perform useful work. In other words, we build objects to do discrete tasks within an application. We then build processes to mimic the human operator's use of applications by calling on the objects we have built.

So as we said, a process uses parts of a business object to manipulate an application and we will soon see that the object studio looks very similar to the process studio. There are however some differences that we'll look at shortly, but the key points to note at this stage are that the object studio offers some types of stages not available in the process studio and that the object studio does not have a main page, but two default pages instead.

Additionally, pages within an object are organised in a flat group rather than in a hierarchy as we saw when discussing process layers. This means that where a business process flows from a single start point on the main page, the flow through a business object can start on any page and will complete that page and stay there until it is told where to go next.

Now let's talk about navigating the object studios using an example. Let's head over to the Blue Prism and select the studio tab.

In this exercise I'll show you a Business object I've created earlier and we'll use this business object in conjunction with Action Stages in a process studio to do some work in Microsoft Paint.

So let's jump into the object studio and I'll open my MS Paint Object. You can see in this Object I have two default pages, Initialise and Clean Up, as well as three pages I've created.

The first page is pretty simple and it launches paint. To show you this, I can simply click reset and press go. As you can see, paint has been launched.

Now, if I go into my next page titled open image, I can see it's a bit more complicated. In this page the object will press all the buttons required on our application (in this case Microsoft Paint), to open an image. First it will click the file button, it will click the open button, we look in our downloads folder, click the image we want and then click open.

We'll adjust it a bit better so you can see the image, and there you go.

Finally we have our close page which you won't be surprised to learn, closes paint. If we reset again, press go we'll see paint has been closed.

Now, we've stepped through this object manually but how do we incorporate this application functionality into a business process?

We'll need what we call an Action stage, so let's learn about action stages and how we use action stages within a process to call on objects that then interact with applications as we've discussed before.

So as we've seen, we have a basic business object that can lunch Microsoft Paint, open an image and close, however, we also know that a business object can only be used via a process, so we'll need to return to the process studio to employ our business object.

Action Stages

An action stage allows us to call on a business object in a process. These action stages will allow the flow of our process to go down into the object layer, interact with the application as specified by our business object and come back up to our process layer when complete.

This flow is similar to what we saw when we discussed process layers, however instead of a page stage sending us to another page within our process, an action stage sends our process to a page within a business object.

Now, let's go into the process studio (note the difference between the process studio and the object studio) and create a new process called Paint Process.

Let's create a process that uses action stages to launch, open an image and close our Microsoft Paint Application.

First, let's click and drag over an action stage, open it up and call it Launch Paint.

Next you will see two drop down menus. The first is where you select your business object and the second is where you select the action, or the specific page you want to use within that business object. You'll notice that until you select a business object you won't be given the option to select an action. This is because we first need to tell Blue Prism which business object to look at before it can tell us what actions (or pages) are available to us.

So, we want to use our MS Paint Object, and given this is our launch action stage, we want to select the launch action of our object. You'll notice the other options from the object we looked at earlier in this lesson. No we click ok.

So now we add another action stage and we'll call this Open Image. We'll select the MS Paint Object and this time we'll select our Open Image Action. Now ok.

Now, last but not least we'll add another action stage, call it Close and from the MS Paint Object we'll select Close and click ok.

Now let's link it up, and press reset. We'll check there's no errors but before we click go what do you think we're going to see?

Well our process is going to call upon our MS Paint Object to carry out the Launch, Open Image and Close actions. So I’m expecting to see something similar to what we saw in the object studio when we were familiarising ourselves with this object, but instead of us stepping through each page manually our process is going to carry out these steps for us.

Let’s give it a go!

Great! We saw our process open Paint, it opened an image and when once complete it closes paint.

Building an Object

Overview

Hi and welcome to the lesson where we will be building a Business Object in the Object Studio.

In this lesson we'll use what we've learnt so far to create a business object that will interact with the Windows Notepad application. We'll do this in the object studio and we'll also be using the Application Modeller for the first time. At the end of this lesson you will have created your first business object so once again, it's time to turn those excitement levels up.

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.

If you recall back to the previous lessons, we know that the purpose of an object is to interact with an application, which provides us with the means to use that application. In order to create this integration we need to teach the business object about the application we'd like it to interact with, by creating an application model.

Microsoft Notepad

Firstly, if you're not already, familiarise yourself with the Microsoft Notepad application. It is a simple application where you can write text on notes and save notes as required. We will be using Microsoft Notepad as what we call our target application.

You can find it by searching Notepad or navigating to the link on your screen. Note that Notepad is a .exe or an executable file. This will be important to remember in a second.
C:\Windows\System32\notepad.exe

Alright, I can tell you're eager to get started, so let's jump straight into Blue Prism. We'll start by creating an object called Notepad in the Object Studio.

Now, as before we can see we have our two default tabs and an Action one tab. Let's call this tab launch.

We now need to introduce Blue Prism to our Target Application (Microsoft Notepad) so let's open our application modeller by clicking the button at the top of the screen.

Application Modeller

Application Modeler is the tool we use to "teach" Blue Prism how to interact with an application.

The first time you open the application modeller in a new object a dialogue box will appear asking for some initial details. As you can see, the application name is prefilled based on the name of our object but this can be changed as required.

We then select the type of application we're going to be working with. Notepad is a Windows based application so we're happy with this.

We recall that Notepad is an executable file so we click next here as well, and finally we need to find the location or our application.
We click browse and we find the notepad application. As we said, my Notepad is in C drive, the windows folder and the system 32 folder. Yours may be in a different location so make sure you find the correct location of the Notepad executable file on your computer before moving on.

We can now click next for the remaining steps and accept the default values. Great, we've introduced Blue Prism to our Target application, Microsoft Notepad, now we need to help them get properly acquainted. Firstly, let's close this window, reenter the application modeller, and we can see we don't need to worry about the wizard anymore.

Now that we've connected Blue Prism application with our executable file, we can click the launch button in the bottom right corner of the window which will launch the application. Great, you can now see Notepad on our screen after we press launch.

Now that we have the location of the application mapped, we want to teach BP what the Text Window of Notepad looks like. To do this, click on element 1 in the element tree here on the left. We'll change its name to Text Field here, then instead of Launch we press the identify button. You may have noticed that as soon as we launched the application the launch button was replaced with the identify button.

We are now in what is called spy mode. When you move the mouse over the Notepad window you will notice that individual elements within the Notepad application are highlighted with a red box as we move the mouse over them. At the moment we're in Win 32 Spy mode but there are other spy modes that can be selected. In this course we'll only be using Win 32 spy mode but we discuss other modes in detail in the excellence phase of the WYWM Blue Prism Developer training.

Blue Prism spys elements as the mouse is moved over the application window. We can select an element within an application and capture it's details by holding the control key (or the command key on a mac) and clicking our left mouse button. To exit spy more we hold down the control button again but we right click our mouse. As you can see, if we don't hold the control key down we can interact with the application as normal.

In this exercise we want to select the text area of our notepad window. So we move the mouse to the text area until it's highlighted, we hold down control and left click on our mouse.

See that the application modeller is now populated with a range of details that it has collected from the text field of our notepad application. These details are called attributes. Attributes are not created by Blue Prism, rather they are characteristics of the element that was highlighted and they are what Blue Prism uses to differentiate between different elements within a particular application window.

Once captured, these attributes allow a business object to remember an element and it does this by using a selection of attributes as the fingerprint for that element. This fingerprint is then used to identify and interact with the application's element, from the object studio.

When we select an element in spy mode, the Application modeller retrieves all the available attributes from the application window but only some of these attributes are used to form the element's fingerprint.

To inspect which elements Blue Prism has used to create this fingerprint, we can click the match column and the attributes that have a tick next to them are ordered at the top of the page. In this case we can see that Blue Prism has used Window Text, Visible, Screen Visible, Enabled and Class name to identify the text window of our notepad.

These elements can be checked or unchecked as required. The initial selection is the default choice made by Blue Prism when the element is spied, and while this first choice is frequently correct, it will often require some manual adjustment to get right. To make these adjustments we can select and deselect other attributes as we like to make our own unique fingerprint for this element. That said, we need to be careful to select appropriate attributes based on our knowledge of the application so our application modeller doesn't lose sight of the element it's looking for.

For example, if we use the height attribute to identify the text element of our notepad, it would require the window to be the exact same height every time it was used. As you can imagine, this would likely cause problems as soon as we change the size of our application window.

We can test now by ticking the height attribute and we can ask Blue Prism to look for that element by clicking the highlight button. Here we can see that the application modeller can still see the text field of our notepad based on it's attribute fingerprint.

However, if we change the height of our Notepad, and click highlight again, you can see that we'll start to run into problems.

An error is produced because the application was looking for an application window in Notepad that was a specific height. We can see this specific height value in the values column adjacent to our Height attribute. As soon as the height of our window changed and it was no longer the height value we can see here, Blue Prism couldn't see it any more and we got an error message. This highlights the importance of selecting suitable attributes when we're identifying elements within the application modeller.

Now let's deselect height, confirm it's working again by clicking the highlight button and we'll rename this element Text Field. Now click okay and we can start to set up our object.

Navigate Stages

Now that we've linked our Notepad application to this object using our application modeller and we've told our object what the text field of notepad looks like, we can use a navigate stage to navigate our object in the direction we want it to go.

As we saw with our paint process before we start interacting with our application we first need to launch. Now it's linked to our application modeller, we have enough information to do this. So let's drag and drop a navigate stage onto the page and go into the properties window.

We'll call our navigate stage Launch. It's always a good idea to be very prescriptive when naming our stages. As was the case with our processes, this helps remove ambiguity and will assist with troubleshooting down the track.

As well as the name and the description, in the properties window you can also see the element tree that we saw in the application modeller on the left of the window. In the same way that we dragged stages in our process studio, we also drag and drop elements into the element action window and assign them an action.

So let's drag in the Notepad element to the element column. And what action do we want this navigate stage to carry out? We want it to launch so let's select launch from the action drop down. By doing this we're essentially telling Blue Prism to launch the notepad application. Now click ok.

Now let's create a new tab and call it Write on Notepad. This time we'll use a write stage to, you guessed it, write on our notepad. But before we do this we'll need a data item so we can tell Blue Prism what we want to write.

Let's get a data item, make it bigger, call it Blue Prism Message, it'll be a text data type and we'll enter our message. It can be whatever you like.

Also, before we write a message we need to check that there is a place to write something. Here we are getting Blue Prism to check that our text field exists. We do this by using a wait stage. We'll cover wait stages in more detail later in the course but for now we drag a wait stage in, rearrange it a bit and get it to check that the text element exists. We do this by dragging the text element into the element field and setting our condition to check exists.

Now let's drag that write stage across, make it a bit bigger and call it Write on Notepad.

In the same way we did before we'll drag the text that we want to write in the value column, and we drag across the element we want our value to be written to. You can think of this as an expression window in a calculation stage, with the element column representing the store result in field. And look at that, we even have an expression window where we can validate our expression.

Now let's add another tab to terminate notepad once we're finished. How do we do this? Well if you answered, the same way as we launched Notepad, you'd be spot on.

We get a Navigate stage, we make it bigger, we call it terminate, we select the notepad element and our action will be terminate. Now all we need to do is link it up and press reset.

You'll notice there is an error. Don't worry, this is related to our wait stage and we'll learn more about this in the wait stage and exception handling lessons. All we need to know now is that our object won't be affected by this error so let's press on.

Ok, let's give our object a test, but first, note that unlike a Process, our tabs, or Objects Actions, are not linked so each one runs independently. So first let's launch our notepad from the launch tab. Great that worked, now go to the Write tab, reset again and press go.

Great, we can see now that Blue Prism has written us a note. Now, because we're sick of Blue Prism showing off, go to our terminate tab, reset again and press go to terminate our application.

Great, we just built our first object.

The only issue that we have now is that these object won't yet be visible as actions from our process page so we won't yet be able to call on them within a process. To resolve this issue we need publish our object pages of actions.

Publishing an object's action allows us to call on the object we've just built and use it within a process. We do this by simply right clicking on the page tabs within our object and we select publish. Once a page is published these icons appear, denoting that this page will be available from the action dropdown within our action stage in the process studio. Now we remember to click save and exit.

Navigating Processes and Objects

Overview

Hi and welcome to the lesson where we will be looking at navigating processes and objects.

In this lesson we'll consolidate what we've learnt so far and integrate the object we just created within multiple processes. We'll also revisit the concept of process layers and call on subordinate processes under a master process.

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.

So let's get started and this time we'll jump straight in Blue Prism and familiaraise ourselves with our notepad object.

Calling on an Object

We'll open our Notepad object from the object studio. You will recall this object consists of 3 actions; launch notepad, write a note and terminate, or close notepad

So let's build a process in the process studio that calls on these Object actions in the order that we want..

Let's call this process Notepad Process, and start by getting some action stages that call on the pages, or actions within our object.

We'll call our first action stage Launch. Do you remember what we need to do to set this action up with the launch page of our Notepad object?

If you said, select the object we want and select the action within that object you'd be right. Let's do that now.

Now let's call our second action stage Write. We'll select the same object but this time we'll use the Write Action.

Finally, we'll call our last action stage Terminate, we'll select our object again and we'll select the terminate action.

Now let's link it up, reset and press go. As you can see, this works the same as we saw in the object studio but we didn't need to step through each page manually as we did before.

Process Reference Stages

Great, now let's add some complexity using some of the processes we've used before, but first, let's quickly discuss Process Reference Stages and a bit more on process layers.

We've seen before that we can spread a process across a number of pages which are arranged in a hierarchy. So far through, we've assumed that all processes are on the same hierarchical level with page reference stages giving us the ability to add a process layer. This is not always the case, and we often have a hierarchy of processes that begins with a master process and below our master processes we have subordinate processes that are 'marshalled' by the flow of our master process.

When we jump back into Blue Prism, we're going to build a hierarchy of processes that looks like this. As you can see, we're going to revisit some of the processes and objects we've already created and call on them through using process reference stages.

As we'll see shortly, the process reference stage is similar to a page reference stage and allows us to reference processes and incorporate them in a layer below the process in which they're referenced.

Let's jump back in to Blue Prism and create a new process called Master Process. Also, feel free to pause this part of the lesson at any time and refer back to the process diagram that we just had a look at for extra clarity.

First let's add two process reference stages to our master process and make them a little bigger. Our first process reference stage is calling on our Collections and Loops process so let's name it accordingly. All we need to do now is select the process we want our master process to be sent to. Down the track we will need to consider the inputs and outputs required for this process but we'll cover that in more detail in the inputs and outputs lesson later in the course. So we now click ok.

Our next process reference stage will call on our the notepad process that we just made, so let's call it Notepad process. We now select our Notepad process and click ok.

Now let's link everything up.

Great so now we have two process layer and were referencing an object on our second process layer.

Now we reset again but before we press go, what is going to happen here?

Well first we will go from start on our first process layer, down into our collections process, and our collections process runs on the second process layer. Once our collection process is complete it will come back up to our master process which is our first process layer. We then progress to our notepad process reference stage and drop back down to our second process layer.

Here we'll launch, write to and close notepad, and each time an action stage is reached within our notepad process, it will call on that action within our notepad object and effectively jump across into the object studio to run that object. Our notepad process will then hit it's end stage and jump back up to the first layer on our mater process before we reach the final end stage and the whole process is then complete.

Cool, now let's see it in action.

Now that seemed to work but why didn't we see anything? Well, when a process reference stage is reached, Blue Prism actions that stage by going through the process almost instantly in the background.

So how can we walk through each step of this process at every level of our process layer including our object?

Do you remember the step function? Well we can use this button to step through each step Blue Prism takes when executing a process. So let's use this to step through our process and we'll talk through it together.

First we go into the Collections process and we can see that we're processing our employee data as we have before.

Once that's done we jump back up to the master process and move from our collection process reference stage to our Notepad Process reference stage. We now jump back down to our second layer and go through our notepad process.

This time, because we're using action stages to call on an object, we are redirected again to go through the actions within the object we're calling on. Note that the window also changes title from process studio to object studio when we're in our object.

Now we're back in our notepad process, we finish that when we reach the end stage, and jump up again to our master process and end the master process.

Feel free to pause the lesson and run this again using the diagram we used earlier in the lesson.

Let's reset and we'll briefly touch on the step over function and the step out function. We use the step over function to step over a stage. Let's demonstrate this by clicking step again and instead of going down into our collection process, let's step over it. Blue Prism will still carry out this process but it will do it in quick time as we saw before when we press go.

We also have a step out function. This allows us to step out of a process and will take us up to the next process layer. We can demonstrate this by clicking step again and going into our notepad process. Now, instead of clicking step again, if we click step out we are taken back to the master process. By doing this, we’ve travelled up a process layer, and as we saw with the step over button, the remainder of the process we were in has been carried out.

This functionality is handy to illustrate our process flow but it is also used when troubleshooting so we’ll touch on it again later in the course.