Robotic Automation - its process automations where computer software drives existing enterprise application software in the same way that a user does.

It creates a virtual workforce which helps in rapid automation of manual, rules-based, back office administrative process.

It reduces cost, improves accuracy and enables business operation to be agile

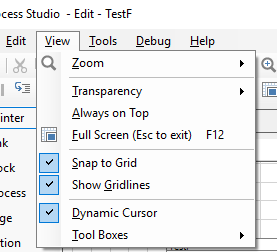
**Process Studio –**

The Blue Prism process is created as a diagram – very similar to business flow diagram created by MS VISIO

It’s a graphical representation of an underlying computer program.

Process diagrams are made up of various ***stages*** connected together using ***links*** to form logical structures.

**Some Process Stage Tips -**

The “grid lines” and “snap” settings are on by default but these can be switched off (via the View menu) if necessary

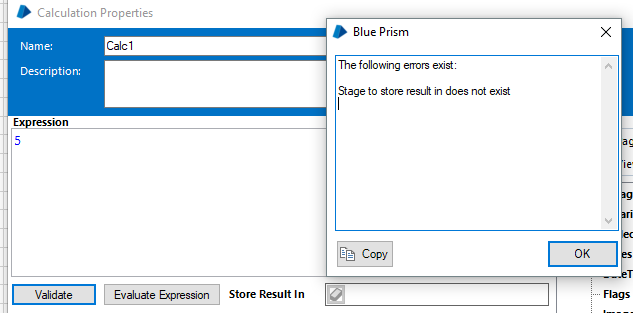
The “Start” and “Information” stages are permanent and cannot be deleted.

 - Process validation button. List basic error and warnings

Get into the habit of using the validation tool as you go along, and in particular, run the validation before you close the Process.

If you don’t use the validation tool you will almost certainly have a glut of silly problems

**Calculation Stage -**

If we don’t provide Store Results In and Click on Validation, It gives us this Error pop up

If we don’t validate here, we can also validate using *Process Validation button* on tool bar of process

F5 is for Run, F8 is for adding a Data Item

Tip: It can help to think of “Store Result In” as meaning “Z becomes 10” rather than “Z equals 10”. Z is the data item

Square brackets must be used to include a Data Item in an expression, e.g., [Account ID]. Calculation stage properties and change the expression to [X]+[Y]

**Break Points -**

The Reset button readies the Process to run again – resetting Data Items to initial values

Although most stages have only one outbound link, there is no limit to the number of inbound links a stage can have. This means that a stage can be approached from more than one direction.

To step though the process one stage at a time click the step button 

Step(~F5)/Step Over(~F6)/ Step Out(~F8). F5 to move from one breakpoint to other**.**

**Step Out will come out of the inner page, also execute the whole process in one go**

Pause the Process and modify the Current Value of one of the Data Items so that the Decision will route the flow to the End stage. This can be done on pausing

**Set Next Stage -** When running a Process, this feature can be very useful for skipping past a section of a diagram. However, be aware that Set Next Stage may have an undesired effect if the new position contains logic, dependent on the section that was skipped over. Set Next Stage is also useful in “replaying” sections of a diagram by jumping back to an earlier position.

Set Next Stage does not “fast-forward” or “rewind” but simply “jumps” forward or back.

Breakpoints only take effect when the diagram is open. In the Production environment, the diagram is not displayed when a Process runs and Breakpoints are ignored.

**Collections -**

A Collection is a type of Data Item similar to an Excel spread sheet. It can hold multiple values like a table with columns and rows.

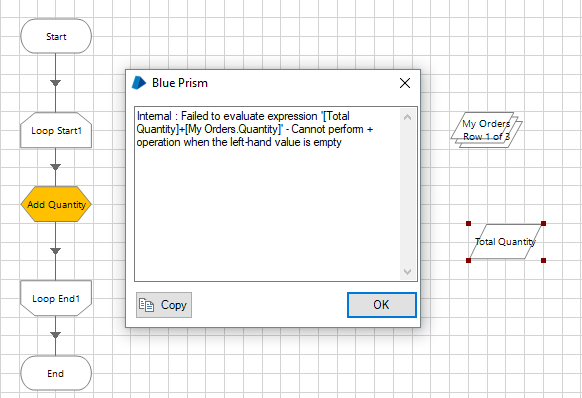
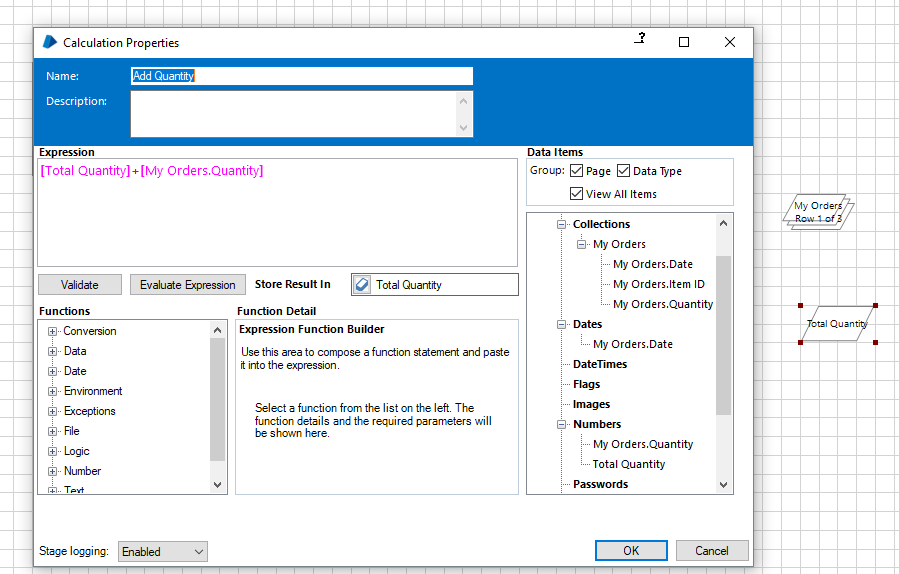
Collection data are accessed one row at a time using a Loop stage to move forward through the rows.

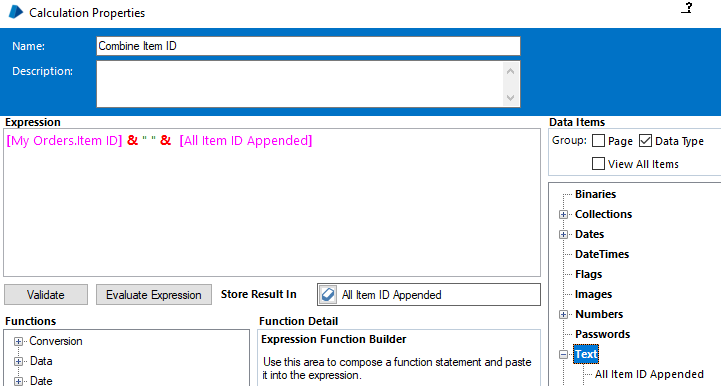
Again, a programming comparison can be made by likening a Collection to an **array** and a Loop stage with a **“for each”** statement.



Dot notation is used to refer to Collection fields, i.e., “Collection Name.Field Name”

Initial value for Numeric Item must be set to some value, 0, else it will throw error (Not needed for text)





Tip: To move to the next row of a Collection (skip the current row), link to the end of the Loop, not the start. Linking to the start resets the Loop to begin from the first row of the Collection.

dateDiff(9, fromDate, toDate) will be greater than 0 if “To date” greater than “From date”. (9 – diff in days)



**Deleting a Page -**

Pages can be deleted if necessary **but care must be taken with any references to a deleted page**. If you rename a page, the references automatically get updated

When you come to develop “real-life” Processes, you will find diagrams can become large and unwieldy, and separating the logic out on to more pages will help to keep your diagrams tidy and readable

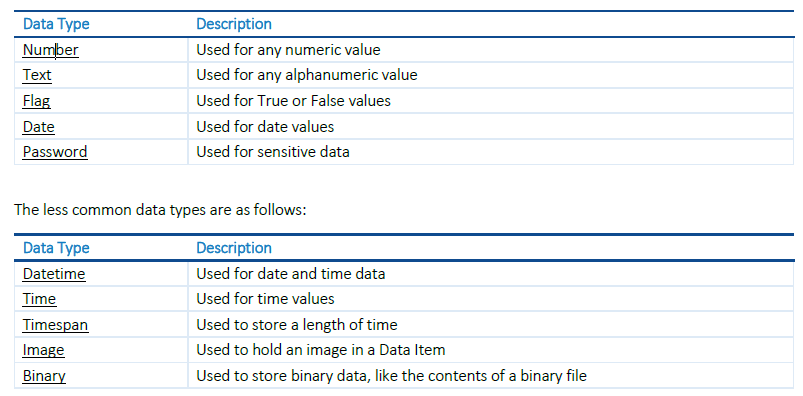
You will almost certainly not be working alone on a Blue Prism solution and, as with any shared document; it makes sense to keep things clear and intelligible to others

The exception is the Main Page, which cannot be referenced

**Using Global Data Items –**

Keeping track of the current value of a global, particularly when working one case after another, needs careful attention. A global can present the risk of a value from a previous case affecting the next case – something that needs to be addressed in the design of a Process

**Data Types - Ten** data types to choose from



The data type you will probably use most often will be Text, particularly when reading from files or an application. Some values may genuinely be Text, e.g., “Smith”, and some may be other types of data presented as Text, e.g., “09/17/1982”. Although it is possible to transform a value from one data type to another (and we will look at casting later on), consideration should be given as to **whether it is actually necessary to do so.**

For example, if a telephone number was read from an application as Text, the Data Item value would be an exact copy of the value in the application. However, if the value was read as a Number, the “leading zero” would be truncated.

AddDays(Today(), [Number Of Days]) -- > Date

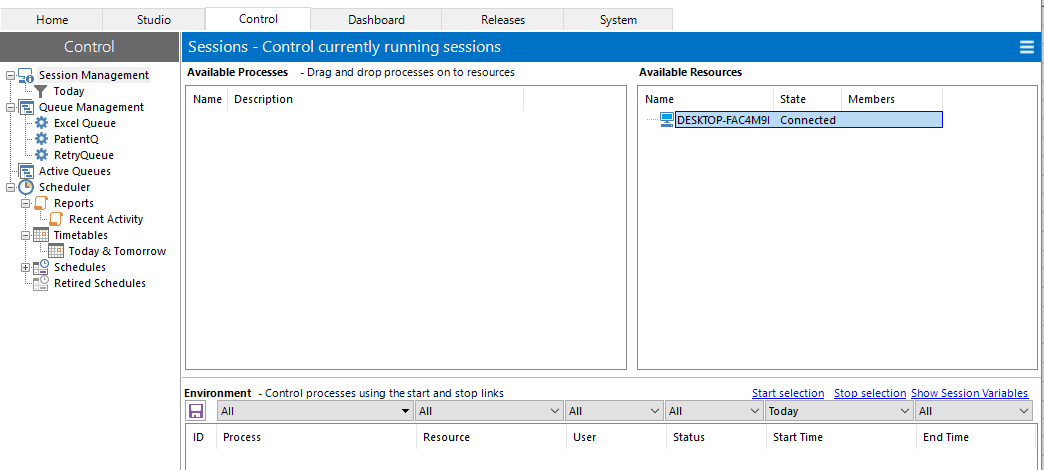
**Start-up Parameters –**

Inputs can also be applied to the Process itself, enabling a Process to accept external values when it starts to run. Process inputs are known as **Start-up parameters**.

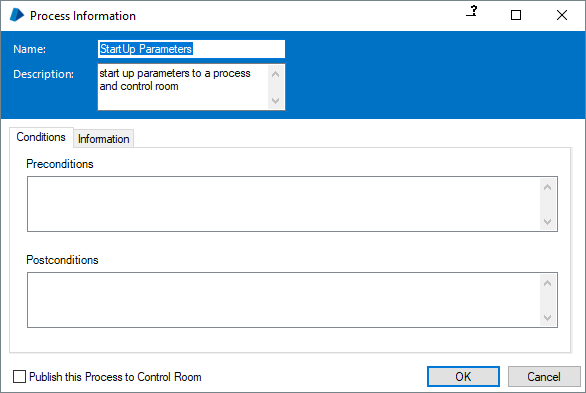
Add input parameter to start stage of main page and store the value in a data item

**Control Room –**

**Process Studio is only used for building and testing** a Process. Although we can run a Process from the diagram, in an operational environment, Processes are not run in Process Studio; **Processes are run from another part of Blue Prism named Control Room**.



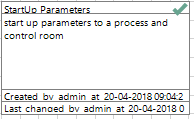
**Publishing -**

By default a Process is not immediately available in Control Room. This is safety feature in Blue Prism to prevent unfinished Processes being run accidently

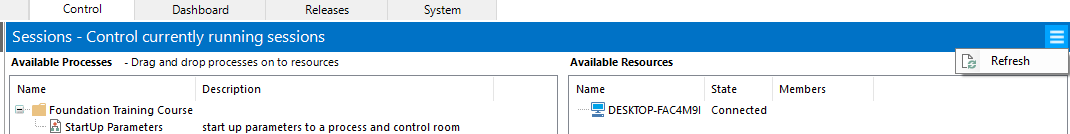
Open the properties of the **Process Information stage** (the large box, usually near the Start stage).

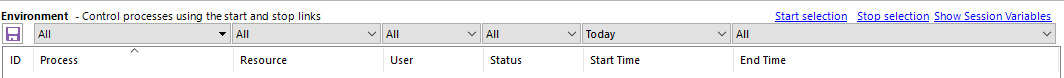
Although for these exercises we are running Processes on the local machine, it is likely that remote machines will be used in a real-world scenario.

**Machines installed with Blue Prism are known as Resources, and an instance of a Process running on a Resource is known as a Session.**

Once published there will be a tick mark on the Process Information Stage.

Refresh button is on the top right of Session Management, three parallel line icon of the control room

****

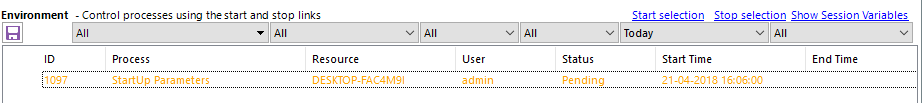


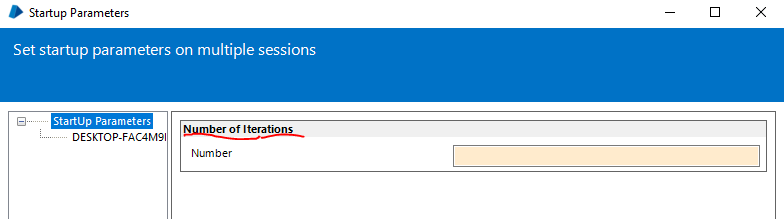
**Running a Session –**

Session Management is divided into three sections –

* Available Processes
* Available Resources
* Environment

When we drag and drop a process onto the resource it shows up in the environment tab at the botton. Since its not started yet, its shown in Orange with Status Pending.



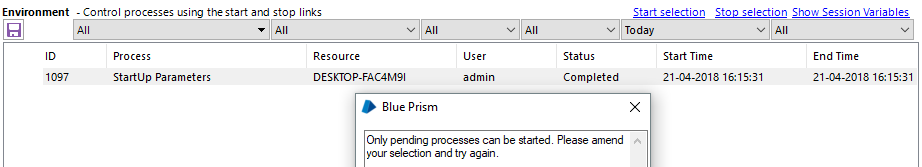


When we click start selection, a pop up will come up asking us for input data

since the start stage in the main page of this process is configured that way.

After the process completion the orange will change to blue to indicate process completion

If we try to start the selection on a session which is already completed, we get the below message



A Session can only run once; to run a Process again a new session must be created.

Orange – Green – Blue/Black

Session status changes to “Terminated” if there is some error in the process and the row becomes black in text

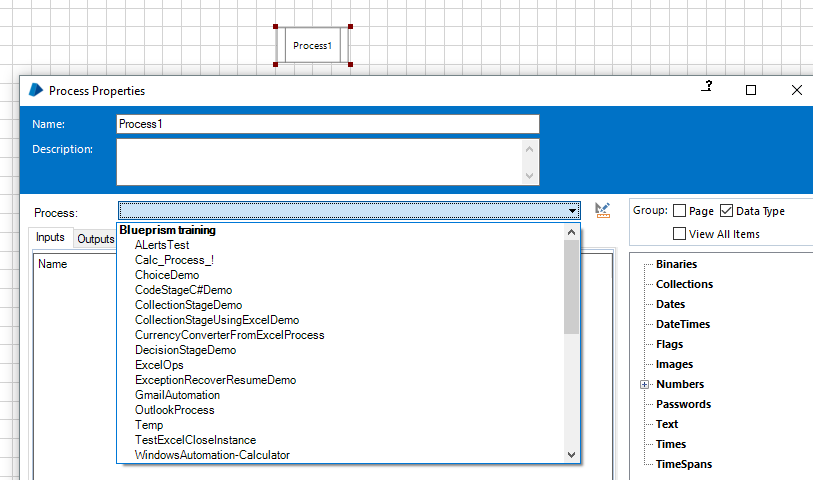
**Session Logs -**

Whenever a Process runs, it makes a record of each step it takes to create a Session Log. You can access this log from Control Room.

* Select one of the Session rows in the Environment list in Control Room.
* Right-click to open the mouse menu and select View Log. You can also double click on a session.

**Process Outputs –**

A process can also call another process similar to calling a page. Process used in this way is often known as a sub-Process



Inputs are used to transmit values from an upper page down to the start of a lower page.

Outputs are used to transmit values up from the end of a lower page back to the upper page.

Start-up parameters can also be used to transmit external values to the start of the Main Page.

**Business Objects –**

A Business object is an interface to **ONE** application.

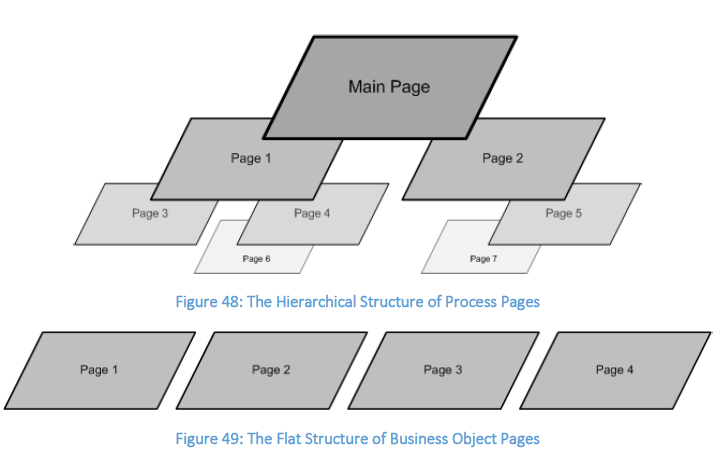
A Business Object is not exposed to Control Room and is never used on its own – it is always used by a Process.

**Object Studio -**

Object Studio is used to **capture the functionality of an application** so that it can be used by Processes.

A Process is used to combine the application functionality with business rules in order to perform useful work.

* Object Studio does not have a Main Page but it does have two default pages. (Initialize and Clean Up)
* Object Studio has additional stages (Read, Write, Navigate and Code, Wait)
* **Pages are organized as a flat group rather than the hierarchy we have seen in Process Studio.**



Business Object pages run one at a time and **are usually not interconnected** like the pages of a Business Process.

A Process can select which Business Object pages to **use and use them in order.**

**Tip: F11 is the Step shortcut key and F10 is Step Over.**

**Notice how Object Studio is opened when you step into each Action stage from Process Studio.**

**Tip: Because Process Studio and Object Studio look so similar, it may not be obvious at first that a new window has opened. You may want to resize or move a window to confirm this.**

Save the Process (NB: you always have to save a Process before running for the first time).

5.6. Review

* Application logic is not contained in a Process.
* A Business Object is used to manipulate an application.
* Object Studio is used to configure Business Objects.
* Object Studio is much like Process Studio.
* A key difference is that a Business Object is not exposed to Control Room; it can only be used via a Process.
* A Business Object encapsulates the functionality of an application and exposes it to a Process.
* A Process uses an Action stage to employ a Business Object page.
* A Process flows through a hierarchy of pages.
* A Business Object flows through pages one at a time.
* Stepping into an Action stage will open Object Studio and stepping out will close it.
* Stepping into a Process Reference stage will open another Process Studio window in a similar way.

Application Modeller that enables us to create a logical representation of an application

The purpose of Application Modeller is to capture details of elements of the application user interface, things like fields, buttons, and windows.

The responsibility for finding the unique combination of attributes ultimately rests with you, the creator of the Business Object.

The application model should not be created in one go but built up gradually. After spying a few elements, return to Object Studio to check their use from the diagram.

Be descriptive when naming a Business Object page – use a ***verb*** to communicate the ***purpose*** of the page. For example, “***Open*** Account History” or “***Close*** Account History” is much better than just “Main Menu”.

Open the Page Information box and provide a description. As you become more proficient you should also document your pre- and post-conditions.

Just as in Process Studio, a Business Object must exist (i.e., be saved) in the Blue Prism database before it can run.

You don’t need to save every time, but you need to save before the very first run.

**Wait Stage –**

It may seem that Order System takes the same amount of time to start up every time. However, in a real-life situation where we would have no control over the response time of an application, and it is not unthinkable that performance could vary.

A Wait stage can be stretched out to accommodate lots of different conditions, providing separate paths to take according to each condition.



When multiple conditions are used, the Wait stage will check each condition in top-down order and opt for the first “true” condition it finds. Polls all conditions simultaneously and goes to the one which is found first

By using a Wait stage without any condition, we can use the timeout as a positive outcome that simply delays the flow slightly. Pauses like this are often used to afford some “breathing space” to the target application, particularly if the application is better suited to a more “human-like” operating tempo.

Be generous with timeouts - overestimate rather than underestimate. Because Blue Prism responds immediately once the expected condition is met

Use Data Items to store timeout values. So that you can modify multiple Wait stages with one easy change to a Data Item.

 Business Objects should have global timeout period data items that are used in all wait stages in the object.

 Using global data items to store timeout periods makes it easy to re-configure how long an application should wait for different types of system activity.

Visit the Initialize page of your Business Object. By convention, this is where global Data Items tend to be created. Create some global number Data Items like those below.

|  |  |
| --- | --- |
| **Name** | **Initial Value** |
| Global Timeout – S | 5 |
| Global Timeout – M | 10 |
| Global Timeout – L | 30 |
| Global Timeout – XL | 60 |
| Global Timeout – XS | 1 |

Typically a larger timeout would be used when

1) waiting for a system to launch,

2) navigate to a new page,

3) update a record, or anything that may take considerable time to execute.

Smaller values are more likely to be used for the more rapid movements of an application, such as

1) waiting for a field to become enabled or

2) a validation message to appear.

**Throttling -**

A Wait stage without any wait condition can be used to create pause, and a Data Item can be used to control the length of the timeout

The “Throttle” global Data Item can be used to manipulate the pace of a Business Object. To test its effect on a target application, a Business Object can be tested with a “tight” throttle that restricts it to the same pace as a human user.

As testing progresses, the throttle can be released until an ideal speed is found. If the throttle is set to zero, the timeout will be instant and there will be no pause.



**Terminate –** is similar to killing an application using Task Manager.

As the name suggests, a termination is rather a severe way to close an application and often the application requires that the user (i.e., Blue Prism) should log out properly beforehand

 Tip: You cannot use a wait to check if Order System has closed because a wait can only work when the target application is running. Once the Navigate stage has terminated, the application any subsequent attempt to use an element from Application Modeller will cause an error.

Use Ordinal attribute (Windows app) to uniquely identify elements when multiple elements are matched



Although Ordinal is a fairly common attribute and can, as in this example, make all the difference, it is not available in all application types.

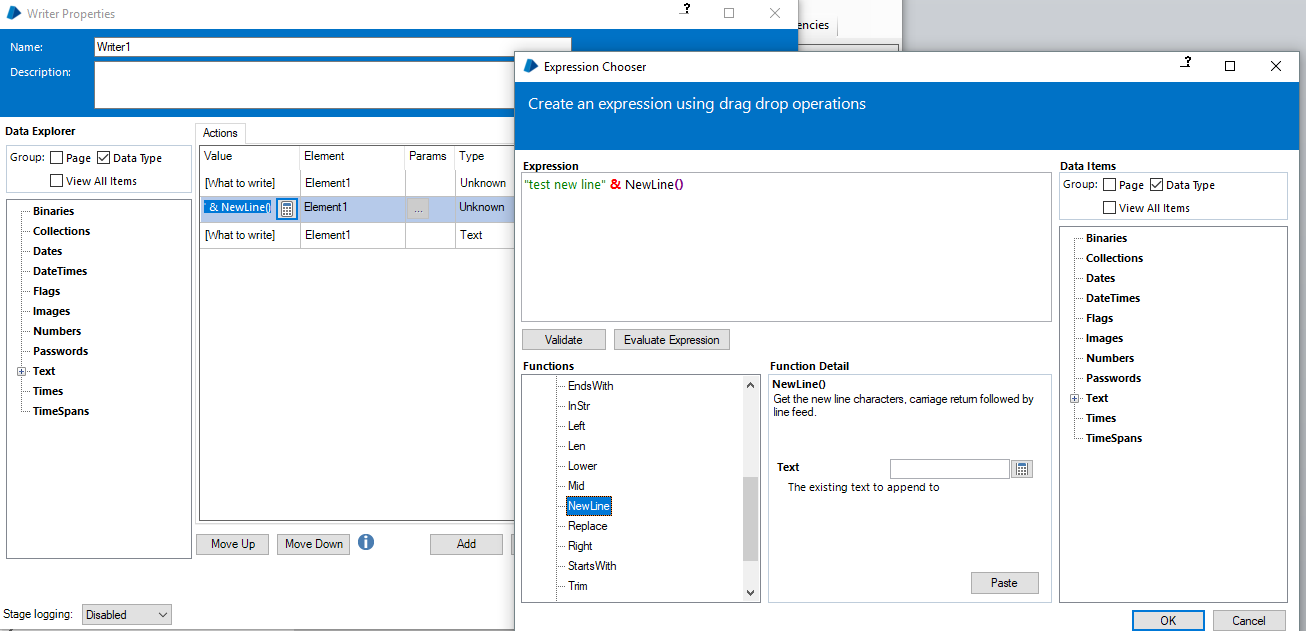
The **“unable to match with query terms”** message is Application Modeller’s way of telling you it cannot find the element. Essentially it is saying, “I can’t see the Staff Number element anymore.”

The selection of attributes can be reduced in order to make an element “correct”. For eg . by default “Window Text=Blank” is checked, on populating the input field, it may fail to read as the text has changed

**Write –**

Tip: You can either create a second Write stage or modify the original Write stage by adding an extra row to the list of Actions.

(If we use the same element across all rows like below, it will just overwrite and enter data from the last row. Will not append)

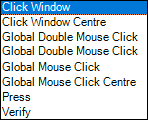
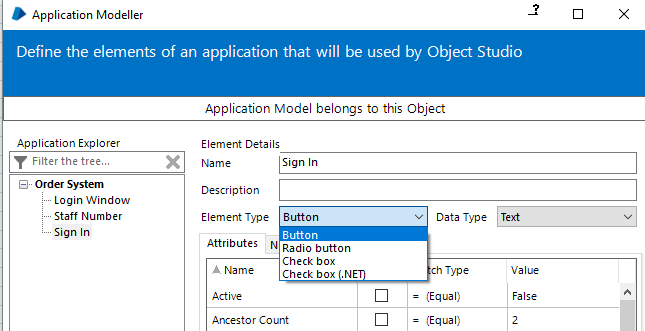


**NewLine()** Text Function to add a line break

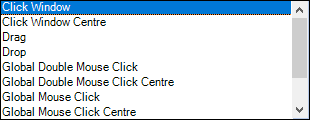
*Tip: In case you were wondering why the field values are overwritten and not appended, the Write stage doesn’t “type” into fields but “gives” the application a value to put into the field. This is also why the field can be populated even if Order System window is covered by another window.*

**Press –**

**The following options will be present in Navigate stage if the element is a button**

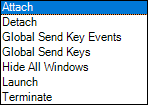
****

**The following options will be present in Navigate stage if the element is an Edit**

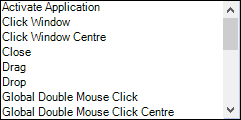


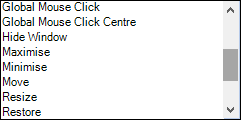
****

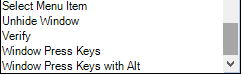
**The following options will be present in Navigate stage for Root Element (Application)**



**The following options will be present in Navigate stage if the element is a window (outermost)**

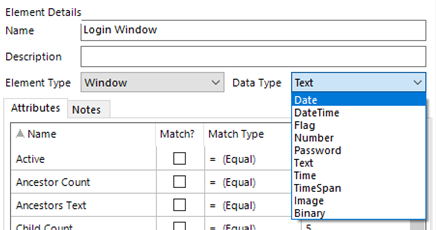


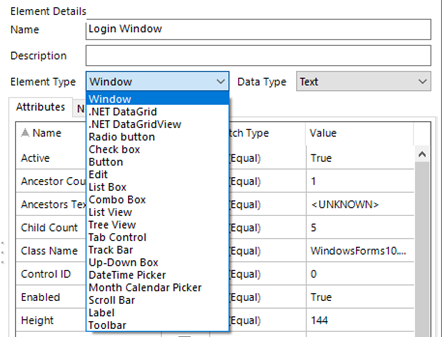


****

**Element Types and Data types.**

On the Application Modeller screen, there is Element Type Drop down and Data Type drop down

****

****

**How to check Checkbox is checked?**

Read Stage - > checked (T/F),

**Read “Application Manager Operations.htm” for detailed mapping of Element Type vs Actions**