

Blue Prism Accredited Developer Exam

Blue Prism AD01

Version Demo

Total Demo Questions: 32

Total Premium Questions: 322

Buy Premium PDF

<https://passqueen.com>

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Process Studio	83
Topic 2, Object Studio	42
Topic 3, Application Modeller	33
Topic 4, Data Types	17
Topic 5, Exception Handling	58
Topic 6, Work Queues	39
Topic 7, Control Room	30
Topic 8, Debugging	11
Topic 9, Mix Questions	9
Total	322

QUESTION NO: 1

When using Application Modeller, which stage is normally used to retrieve the value displayed in an identified application element?

- A. Read stage
- B. Write stage
- C. Decision stage
- D. Exception stage

ANSWER: A

Explanation:

A **Read stage** is normally used to retrieve information from an identified application element. The developer selects the required element and configures the attribute or value to read, then maps the result to an output Data Item.

For example, a Read stage may obtain the text displayed in a customer-name field, the selected value of a list, the contents of a table, or the existence state of an application element. A Navigate stage is used to perform actions such as clicking, selecting, pressing keys, or opening a window, while a Write stage is used to place a value into an element.

See the [Blue Prism documentation portal](#) for Application Modeller and Object Studio stage guidance.

QUESTION NO: 2

Which of the following options can be used to update the value of a data item exposed as an Environment Variable?

(Select all items which apply)

- A. Using the 'Internal - Global Variables' Business Object to update the value
- B. Using a Calculation stage in the same Process to update the value
- C. Using the Global Variables store in Environment Manager to update the value
- D. Using System Manager to update the value

ANSWER: A C D

Explanation:

An Environment Variable is a centrally maintained value that can be exposed for use across Blue Prism processes and business objects. It is appropriate to update such a value through the platform's dedicated global-variable and environment-management mechanisms. The **Internal - Global Variables** Business Object provides programmatic access to global values, allowing an automation to set a variable at runtime when the executing account has the required permissions. This supports controlled, reusable updates from within a process.

The **Global Variables store in Environment Manager** is also designed for maintaining these centrally held values. It provides the administrative interface for managing the configured global or environment-level data used by automations. **System Manager** likewise provides the administrative capability to update Environment Variable values directly in the Blue Prism environment. These mechanisms update the centrally managed value rather than merely changing a process-local instance of a data item.

In production, updates should be governed by appropriate security roles and change-control practices, especially where values contain configuration settings or credentials. Blue Prism documentation is available through the [Blue Prism Documentation portal](#) and the [Blue Prism Customer Portal](#).

QUESTION NO: 3

What are the TRUE statements about a Blue Prism Business Object?

Select the TWO correct responses.

- A. Defined within the Object layer
- B. Will always have a Launch action
- C. Can be used as a stand-alone process
- D. Do not share any stages with the Process Studio
- E. Actions are built to no specific hierarchy

ANSWER: A E

Explanation:

“Defined within the Object layer” is correct because a Blue Prism Business Object is developed in Object Studio, which forms the Object layer of a Blue Prism solution. This layer encapsulates the interface and operations needed to work with a target application or technology. A process in Process Studio can then call the object’s published actions, separating business workflow from application-specific automation details. This design supports reuse, maintainability, and a clearer division between process logic and application integration.

“Actions are built to no specific hierarchy” is also correct. A Business Object contains one or more action pages, each designed to provide a discrete, callable capability, such as logging in, searching, retrieving data, or submitting an update. These actions are not required to conform to a parent-child execution hierarchy within the object. Instead, a process or another object action invokes the required published action as needed. Object actions can themselves call other actions where appropriate, but their structure is driven by the automation design rather than a mandatory hierarchy.

This Object Studio model enables a single Business Object to expose a coherent set of reusable operations for the application it represents. See the Blue Prism documentation home for Object Studio and process-design guidance at [Blue Prism Documentation](#) and the official learning resources at [Blue Prism University](#).

QUESTION NO: 4

According to Blue Prism best practice which of the following information would be appropriate to store in a Work Queue tag? (select 3 responses)

- A. Something that identifies the type of work that was performed Example tags might be: Funds Transferred, SIM Card Requested, Account, Closed, Account Suspended, Account Noted
- B. Something that identifies exact datetime of a step in the process. Example of tags might include Account Closed 20/03/2017 12:12:16, WorkCompleted 20/03/2017 12:12:16: Request Loaded 20/03/2017 12:12:16
- C. Something that identifies the source of work. Example of tags might include: Email, Web Form, Excel Adhoc. Database
- D. Something that identifies the client record to make work queue items easier to search. Example tags might be a phone number, customer reference number or ID, customer full name. For example: +4407994446767, 00023231 A; Mr John A Smith
- E. Something that identifies the type of process scenario. Example of tags might include: Invalid Product, Unrecognised customer, Goneaway Marker, Invalid Status
- F. Details of an exception that has occurred. For example: The Login Screen was not found.

ANSWER: A C E

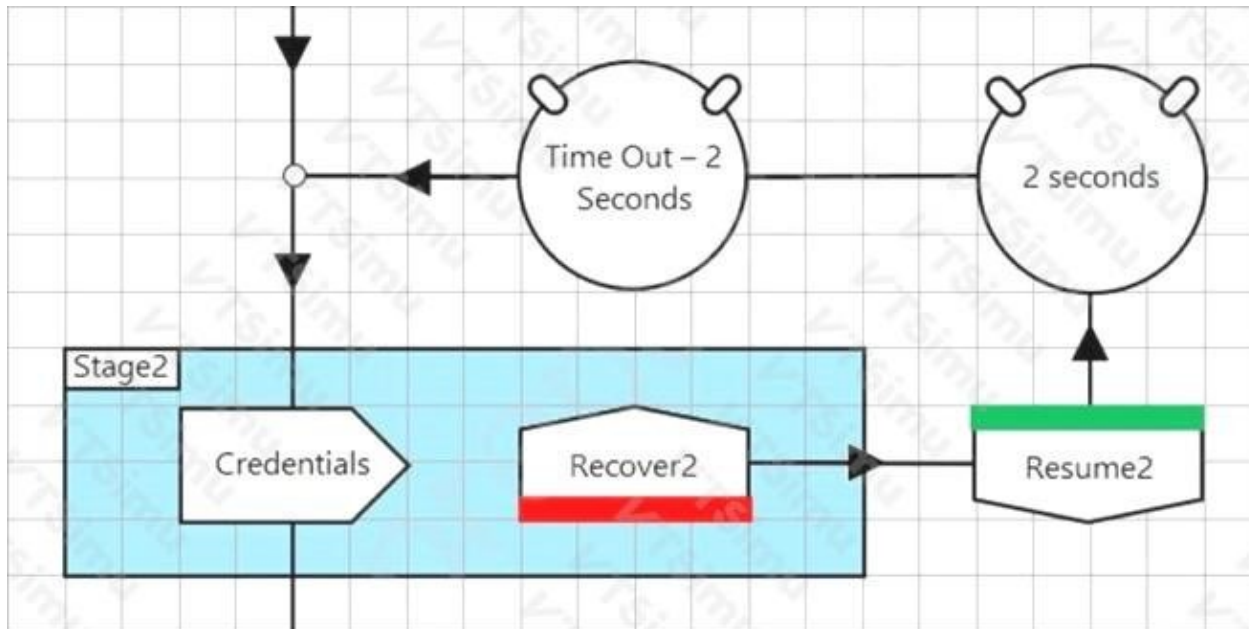
Explanation:

Work Queue tags are intended to provide concise, reusable classification metadata that supports operational reporting, filtering, searching, and analysis of queue items. “Something that identifies the type of work that was performed” is appropriate because outcome or activity categories, such as a transfer, request, closure, suspension, or note, can be

consistently applied across many items and then used to understand the work handled by the process. “Something that identifies the source of work” is also suitable because stable source categories such as email, web form, spreadsheet, or database help teams report on intake channels and investigate trends. “Something that identifies the type of process scenario” is appropriate for the same reason: scenario classifications such as invalid product, unrecognised customer, or invalid status provide a controlled way to group items by the business condition encountered. These tag values are meaningful at an aggregate level and can be applied consistently without making tags overly specific. Blue Prism Work Queues support tagging so that items can be categorized and retrieved according to useful business and process metadata. See the official [Blue Prism documentation portal](#) and Blue Prism’s [guides and resources](#) for Work Queue and solution-design guidance.

QUESTION NO: 5

Examine the following Blue Prism Action taken from Object Studio:



During testing of this Action, the Developer discovered that when attempting to write the target application credentials to the relevant fields in the application, it sometimes failed with an Exception even though the elements existed on the screen.

Though trial and error, the Developer found that they could overcome this problem by simply catching the Exception, waiting 2 seconds and then retrying.

What potential issue do you see with the current configuration of this Action?

- A. In the current configuration, the Resume stage is not within the same Exception Block as the Recover stage and therefore will not work
- B. There is a potential in the current configuration for an infinite loop, if the 'Credentials' stage keeps throwing an exception for some unexpected reason
- C. In the current configuration, the 'Credentials' stage is a Read stage and should be a Write stage
- D. The current configuration includes a Wait stage which is not required
- E. There is nothing wrong with the current configuration

ANSWER: B

Explanation:

There is a potential in the current configuration for an infinite loop, if the 'Credentials' stage keeps throwing an exception for some unexpected reason. The recovery path catches an exception, waits for two seconds, resumes execution, and returns to retry the same Credentials stage. This can be useful for a genuinely transient timing issue, such as an application field not yet being ready for input. However, the retry path has no attempt counter, timeout, or decision point that limits the number of

retries. If the failure is persistent—for example, because the application is unavailable, the element has changed, the session has expired, or the supplied credentials cause an application-side problem—the action will continually recover, wait, and retry rather than reporting a meaningful exception to its caller. Blue Prism exception handling uses Recover and Resume stages to control where execution continues after an exception, so a retry design should explicitly track and cap retries before rethrowing or escalating the exception. This prevents a process from becoming stuck indefinitely and supports clearer operational diagnostics. See Blue Prism’s guidance on [exception handling](#) and its documentation for [Decision stages](#), which can be used with a retry counter to enforce a bounded retry policy.

QUESTION NO: 6

Which of the following statements regarding the use of Work Queues are correct? (select 2 responses)

- A. Control Room allows filtering of the displayed queue items using exact tags and by using wildcards
- B. Wildcards are not allowed in the Tag Fitter property of the Work Queues business object Get Next Item action
- C. Work Queue item Status can only be set to Completed or Exception
- D. Tags can be used allow flexibility in the order items are worked within a single Work Queue

ANSWER: A D

Explanation:

Control Room allows filtering of the displayed queue items using exact tags and by using wildcards is correct because tags provide a searchable classification mechanism for queue items. In Control Room, an operator can use the queue-item tag filter to narrow the displayed items to a particular tag value or a matching tag pattern. This is particularly useful when a queue contains items for several business streams, regions, or processing categories.

Tags can be used allow flexibility in the order items are worked within a single Work Queue is also correct. A process can use a tag filter when requesting its next item, allowing it to retrieve a suitable subset of pending work from the same queue. For example, different workers can request items bearing tags associated with their capabilities or assigned workload. This supports controlled routing and prioritization within one shared queue without requiring a separate queue for every category. Queue-item tags are therefore a practical mechanism for influencing which eligible items a process retrieves while retaining centralized queue management. See the official [Blue Prism documentation portal](#) and Blue Prism’s [product documentation resources](#) for Work Queue guidance.

QUESTION NO: 7

Which of the following are appropriate uses of a Blue Prism Loop stage? (Choose all that apply.)

- A. Iterating through each row in a collection
- B. Processing each record returned from a database query
- C. Calling a published Business Object action
- D. Raising a Business Exception

ANSWER: A B

Explanation:

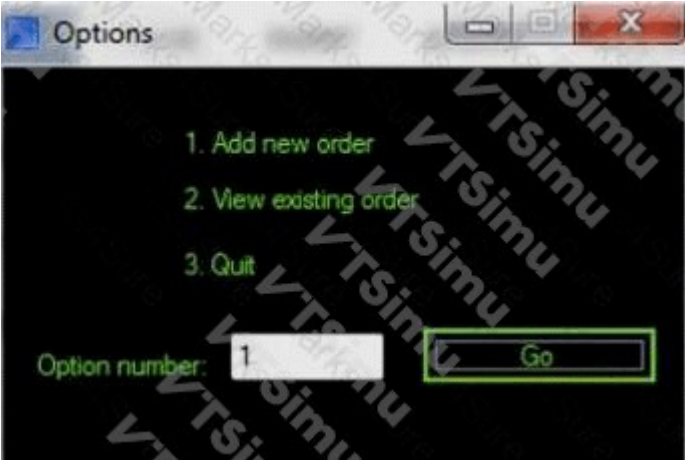
A Loop stage is used to iterate through the rows of a collection. It establishes a current row while the loop is active, allowing expressions such as **[MyCollection.MyField]** to refer to field values from each row in turn.

A Loop stage can also be used to repeat a defined sequence of stages for every record returned from a file, database query, or other collection-producing action. The processing stages should be placed inside the loop path so that they execute once per row.

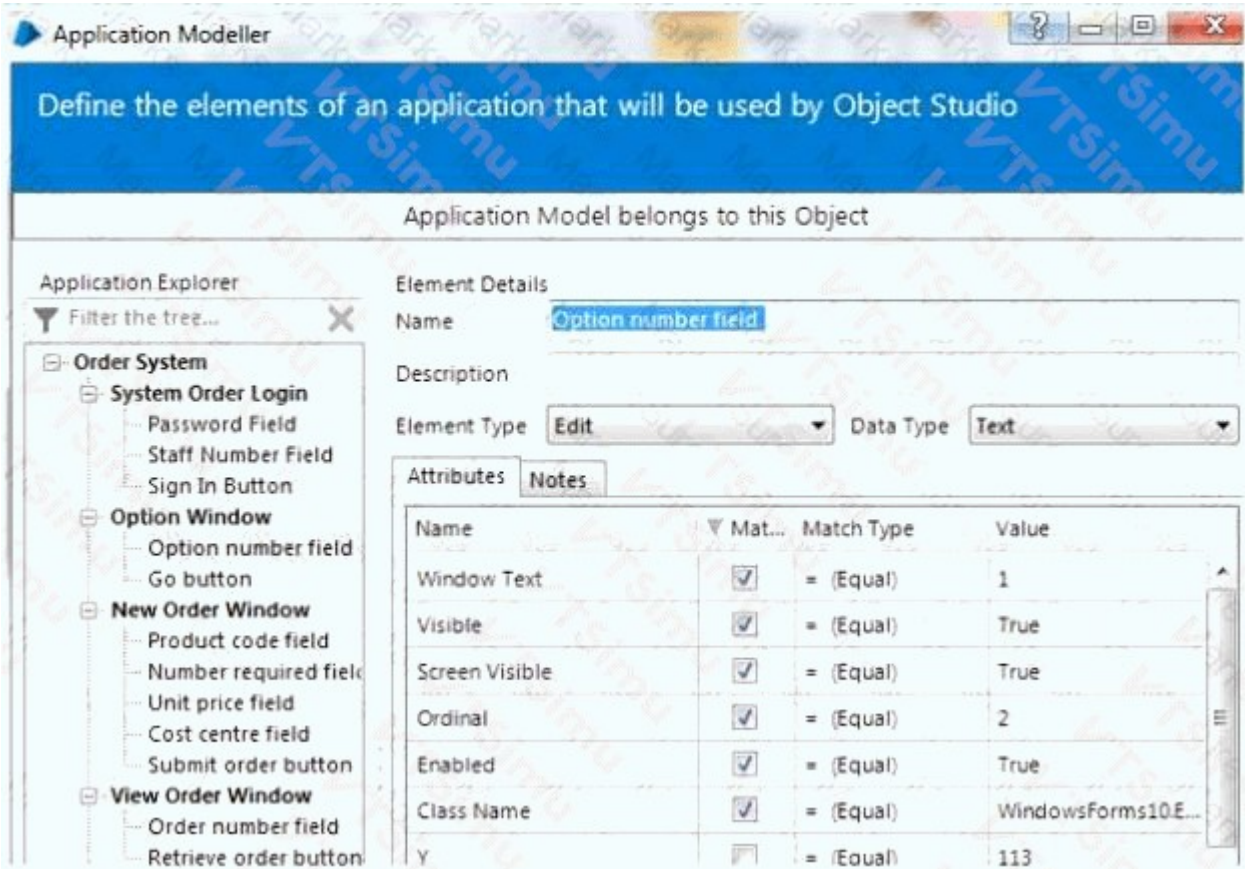
A Loop stage is not used to call a Business Object action or to raise an exception. Those activities are performed using Action stages and Exception stages respectively. See the [Blue Prism documentation portal](#) for guidance on collections and Process Studio stages.

QUESTION NO: 8

When entering an option within the Training Order system the relevant option number is entered into the Option number field as below The Option number input field has been spied in Blue Prism:



The following attribute set has been returned by Application Modeller:



Which Attribute can be un-ticked to ensure consistent visibility of the element to Blue Prism?

- A. Window Text
- B. Visible
- C. Screen Visible

ANSWER: A

Explanation:

Window Text is the appropriate attribute to un-tick. In an Application Modeller definition, selected attributes are used to identify a target element at runtime. The text associated with an input control can change when a user enters a different option number. If that changing value is retained as a matching attribute, the saved element definition may only match the particular value present when the element was originally spied, rather than the input field itself. Removing Window Text makes the match independent of the data currently held in the field and therefore allows the element to remain consistently identifiable as different option numbers are entered. Stable structural attributes supplied by the application, such as the control class and other fixed identifying properties, should be used to identify the control where available. This follows Blue Prism's general approach of selecting reliable, non-variable attributes for an application element and avoiding attributes whose values are expected to change during a process run. See Blue Prism's [Application Modeller documentation](#) and its [guidance on spying application elements](#).

QUESTION NO: 9

Which of the following Work Queue icons indicates a case is waiting to be worked?



- A. Option A
- B. Option B
- C. Option C
- D. Option D

ANSWER: C

Explanation:

Option C is correct because it represents the *Pending* Work Queue item status. In Blue Prism, a queue item that has been added to a work queue but has not yet been acquired by a running process is pending; in practical terms, it is waiting to be worked. When a process requests the next available item, Blue Prism selects an eligible pending item and locks it for that session. The item then remains locked while the process performs its work, after which the process should mark it as completed, exceptioned, or defer it as appropriate. Understanding the pending icon is important when monitoring queue volumes and determining whether work is available for a process to pick up. Blue Prism's queue-management guidance describes the lifecycle of queue items, including their status while awaiting processing, in the official documentation: [Blue Prism product documentation](#) and [Blue Prism guides and resources](#).

QUESTION NO: 10

A Blue Prism solution has been developed, however, the Process cannot be run in Control Room as it is not listed.

Which of the following items may suggest why this is?

- A. The process requires objects that no longer exist or haven't been imported
- B. The process has not been published
- C. Process validation has highlighted multiple errors in the Process design
- D. The process uses objects with different run modes

ANSWER: B

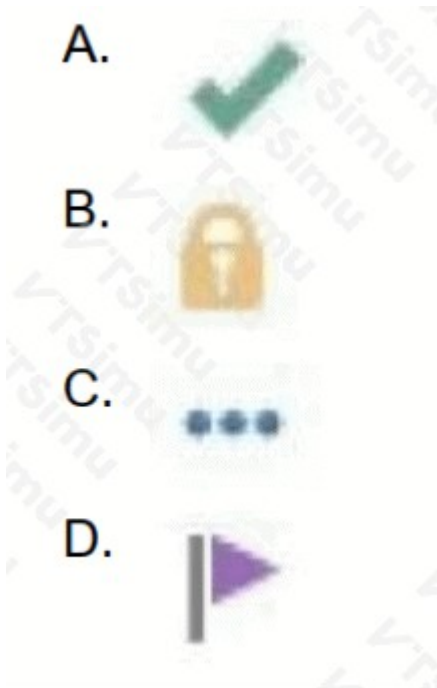
Explanation:

The process has not been published is correct because publishing is the step that makes a process available for use through Control Room. During development, a process can be saved and tested in Process Studio without being exposed as an executable process for schedulers, resource allocation, or manual execution in Control Room. The developer must publish the process from Process Studio once it is in a suitable state to run. After publication, Control Room can display the process and users with the necessary permissions can create sessions for it on available runtime resources.

Publishing also establishes the version of the process that Blue Prism will execute operationally. If later edits are made in Process Studio, those changes do not become the operational version until the process is published again. Therefore, when a developed process is absent from the Control Room process list, checking its publication status is an appropriate first diagnostic step. Blue Prism's Control Room documentation describes Control Room as the area used to run and manage published processes, while Process Studio documentation covers development and publication activities. See [Blue Prism Control Room documentation](#) and [Blue Prism Process Studio documentation](#).

QUESTION NO: 11

Which of the following indicates a case has been worked successfully in a Work Queue?



- A. Option A
- B. Option B
- C. Option C
- D. Option D

ANSWER: A

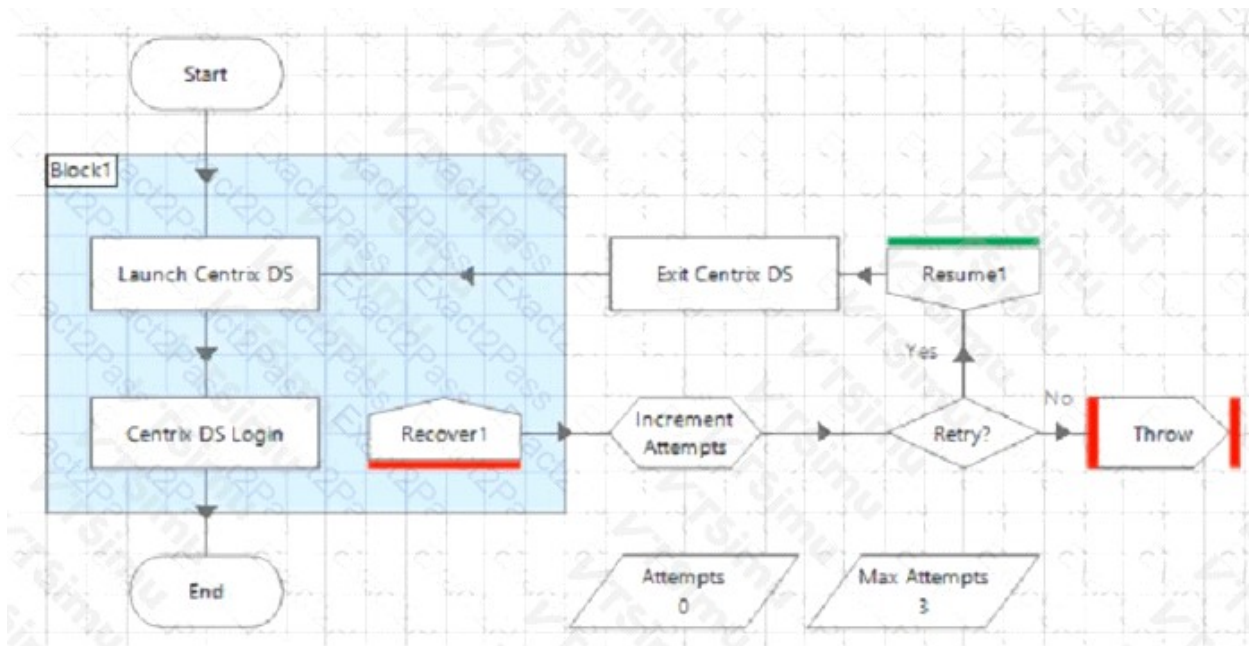
Explanation:

Option A is correct because it represents the successful completion state for a Work Queue item. In Blue Prism, a queue item is initially pending and is locked when a runtime resource retrieves it for processing. Once the process has completed all required work for that item without a business or system exception, it must explicitly mark the item as completed. The completed status is the Control Room indication that the case was processed successfully and no further queue processing is required for it.

This status is significant for operational reporting and queue management: completed items can be distinguished from items still pending, currently locked, or those that require attention because an exception was recorded. A process normally uses the Work Queues business object action that marks the current queue item as completed only after its successful end-to-end handling. Blue Prism documentation describes how queue items progress through these statuses and how Control Room displays and manages them. See the [Blue Prism queue management documentation](#) and the [Blue Prism documentation portal](#).

QUESTION NO: 12 - (DRAG DROP)

Examine the following Blue Prism Process flow:



The Process is intended to attempt to perform the processing enclosed in the recover block (Block1). a maximum number of times before throwing an exception The maximum number of attempts is stored in the ' Max Attempts ' data item.

The ' Attempts ' data item is a number data item with an initial value of 0.

The ' Max Attempts ' data item is a number data item with a current value of 3. This value is provided at start-up.

The ' Increment Attempts ' Calculation stage increases the value of the ' Attempts ' data item by 1.

To enable this Process flow to function correctly, drag and drop the items below into the placement boxes (A - C), to create the correct Blue Prism expression for the Retry? ' Decision stage.



ANSWER:

Answer:



Answer:

Explanation:

A : [Attempts] B : < C : [Max Attempts] So, the final expression should be: **[Attempts] < [Max Attempts]** This will correctly check if the current number of attempts is less than the maximum allowed attempts, enabling the process to retry if necessary.

Explanation:

The Retry? decision needs to test whether another retry is still permitted after an exception has been caught. In this flow, an exception occurring within Block1 transfers control through the Recover stage to the Increment Attempts calculation. This is important because the number held in Attempts represents the failed processing attempt that has just occurred before the

decision is made. The decision must therefore compare the current, incremented value of Attempts with the configured retry limit held in Max Attempts.

The correct Blue Prism decision expression is **[Attempts] < [Max Attempts]**. Data item references in Blue Prism expressions are enclosed in square brackets, so [Attempts] reads the numeric count and [Max Attempts] reads the numeric limit supplied at start-up. The less-than operator returns a Boolean result, exactly as required by a Decision stage. Blue Prism describes Decision stages as using an expression that determines which path execution follows; see the [Blue Prism Decision stage documentation](#).

With an initial Attempts value of 0 and a maximum of 3, the first caught exception increments Attempts to 1 and the expression is true. The process follows the retry path, resumes execution, restarts the application sequence, and tries the protected processing again. After the second caught exception, Attempts becomes 2 and the expression remains true. After the third caught exception, Attempts becomes 3; it is no longer less than 3, so the decision is false and the flow reaches Throw. This gives exactly three processing attempts in total and prevents any further retry once the specified maximum has been reached. The mapping shown in the answer image correctly constructs this expression.

QUESTION NO: 13

When considering an element's attributes in Application Modeller which of the following statements are true? (select all that apply)

- A. The attributes are provided to Blue Prism by the Target System
- B. The initial selection of attributes made by Blue Prism will always be the optimum attribute set for the element
- C. The list of attributes may vary from application type to application type
- D. Selected attributes are shown as ticks in the Match column of Application Modeller

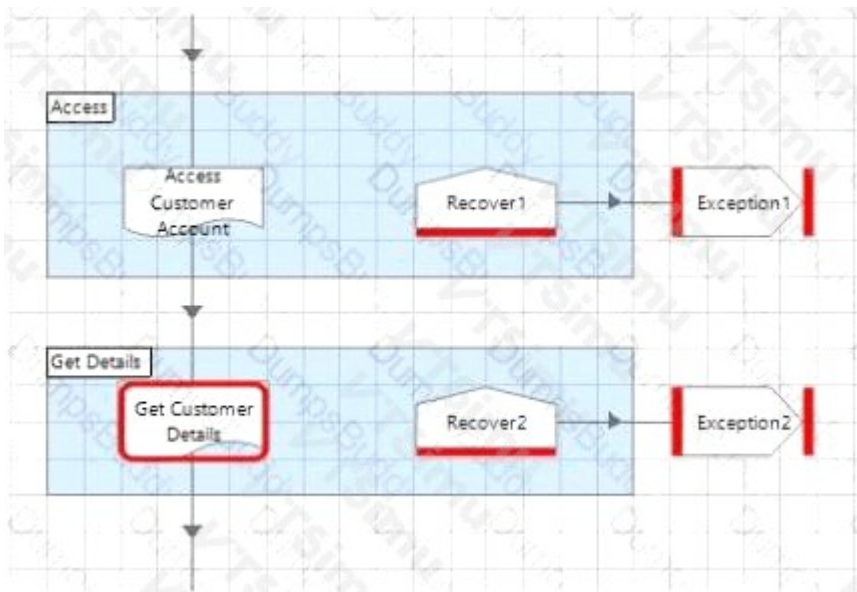
ANSWER: A C D

Explanation:

“The attributes are provided to Blue Prism by the Target System” is correct because Application Modeller obtains the identifying information exposed by the application interface through the selected spying technique. These properties describe the target application’s UI element, such as its name, class, automation identifier, path, or other technology-specific details; they are not a fixed, universal set invented by Blue Prism. “The list of attributes may vary from application type to application type” is also correct. The attributes that can be read depend on the application technology and interface used—for example, a Windows application, browser-based application, Java application, or mainframe terminal can expose different properties and therefore different matching choices. Finally, “Selected attributes are shown as ticks in the Match column of Application Modeller” correctly describes how a developer indicates which available attributes Blue Prism should use when matching the element at runtime. Selecting a reliable combination of these target-provided properties is fundamental to creating robust element definitions and avoiding unnecessary dependence on volatile UI details. Blue Prism’s product documentation describes Application Modeller as the area used to define and configure application elements and their matching information. See the [Blue Prism documentation portal](#) and the [Blue Prism University learning portal](#) for Application Modeller and element-spying guidance.

QUESTION NO: 14

Study at the following diagram which is an extract from a sub-page of a process being tested in Process Studio:



There are no breakpoints in the "Access Customer Account" subpage however there is an error.

What will happen if you choose to step out (SHIFT+ F11) when debugging the process?

- A. The process will work all stages in the "Access Customer Account" page until the error is thrown and then focus would move to the breakpoint at "Get Customer Details" stage.
- B. The process will work all stages in the 'Access Customer Account1' page until the error is thrown and then focus would move to the "Recover 1 stage
- C. The process will work all stages in the "Access Customer Account" page until the error is thrown and then focus would move to the stage containing the error on the "Access Customer Account" page,
- D. The process will work all stages in the "Access Customer Account" page until the error is thrown and then focus would move to the "Exception 1 stage.

ANSWER: D

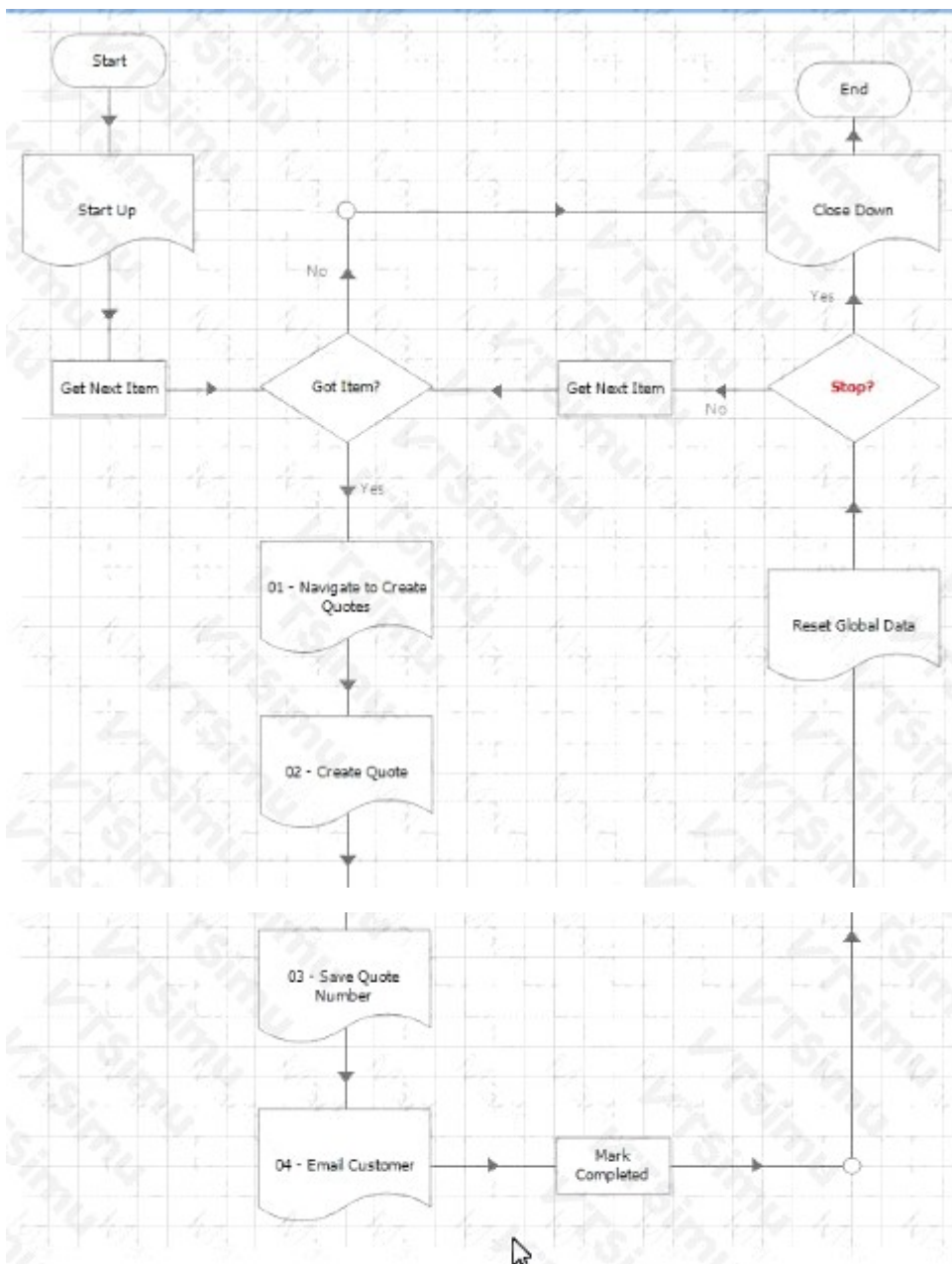
Explanation:

The process will work all stages in the "Access Customer Account" page until the error is thrown and then focus would move to the "Exception 1 stage." This follows the purpose of Step Out in Process Studio debugging: it allows execution to continue out of the currently active called page and back toward its calling context, rather than manually stepping through every remaining stage. In this scenario, execution proceeds in the "Access Customer Account" subpage until the error occurs. The error raises a Blue Prism exception, which is then handled by the exception flow configured in the calling process. Control consequently transfers to the designated exception-handling path, represented in the diagram by the "Exception 1" stage, where the process can record, rethrow, or otherwise manage the exception according to the process design.

This behaviour is important during debugging because it lets a developer quickly observe how an error from a subpage propagates into the parent page's recovery and exception-handling design. It also preserves the normal runtime exception behaviour; Step Out changes how the debugger advances, not how Blue Prism routes an exception. Blue Prism's product documentation provides the Process Studio and debugging guidance used for this behaviour at [Blue Prism Documentation](#), while the official learning portal includes developer training material on process design and exception handling at [Blue Prism University](#).

QUESTION NO: 15

Consider the following process flow from the Main Page of a process in Process Studio:



Thinking about the standard Blue Prism Process Templates, what is wrong with this Main Page process flow?

- A.** There is nothing wrong with the Main Page flow shown in the image, it is a copy of Blue Prism 's standard Process Template
- B.** The Stop? decision stage should be removed from the process flow, as Stop time is Configured either within the Scheduler process or stopped via Control Room.
- C.** The Main Page process flow is too high level. For better visibility of the process logic the flow steps hidden in the sub-pages should all be on the main page instead.
- D.** There is no exception handling on this Main Page so any exceptions that "bubble up" will cause the process to terminate rather than exceptions being handled correctly.
- E.** The Reset Global Data page is not required because sub-page within a process should always use input parameter rather than global data items. There should be no global data to reset.

ANSWER: D

Explanation:

There is no exception handling on this Main Page so any exceptions that “bubble up” will cause the process to terminate rather than exceptions being handled correctly. This is the issue because the Main Page is the final boundary for exceptions raised by called sub-pages or business objects. An exception can be deliberately handled within the page where it occurs, but if it is not handled there, it propagates back through the calling hierarchy. Without a Recover stage and an associated controlled exception-handling path on the Main Page, an unhandled exception reaches the top-level process flow and ends the session without the process being able to perform its intended final actions.

A standard process design uses a top-level exception-handling structure to provide a consistent outcome for unexpected failures. This can include recording useful diagnostic information, completing any required cleanup, ensuring that the process reaches an appropriate end state, and—where required by the design—rethrowing the exception so Control Room accurately records the session as terminated with an exception. This makes failures traceable and predictable rather than allowing a sub-page exception to end execution without the Main Page’s controlled handling logic. Blue Prism’s exception model and the purpose of Recover and Resume stages are described in the [Blue Prism exception handling documentation](#) and the [Process Studio guide](#).

QUESTION NO: 16

Which of the following statements regarding the use of Work Queues are correct? (Choose two.)

- A. Control Room allows filtering of the displayed queue items using exact Tags and by using wildcards
- B. Wildcards are not allowed in the Tag Filter property of the Work Queues Business Object Get Next Item action
- C. Work Queue Item Status can only be set to Completed or Exception
- D. Tags can be used allow flexibility in the order items are worked within a single Work Queue

ANSWER: A D

Explanation:

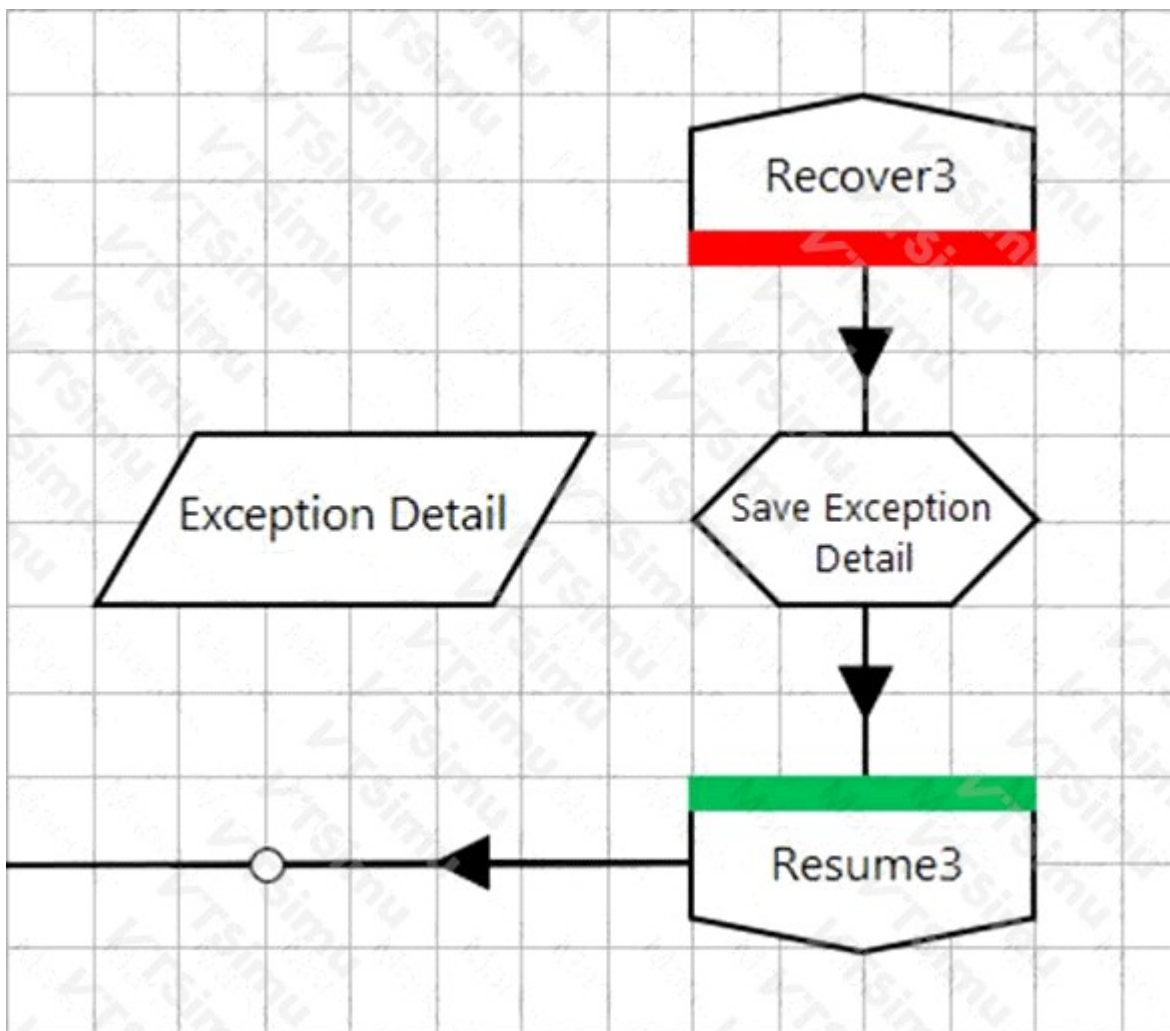
“Control Room allows filtering of the displayed queue items using exact Tags and by using wildcards” is correct because tags provide a practical way to locate relevant queue items in the Control Room interface. When viewing a queue, a user can filter the displayed items using tag values, including wildcard-based matching. This supports operational activities such as finding items assigned to a particular business area, process category, or priority grouping without changing the queue’s underlying data.

“Tags can be used allow flexibility in the order items are worked within a single Work Queue” is also correct. A queue’s normal processing order can be adapted by assigning tags to items and then configuring the Get Next Item operation to retrieve an item that matches the required tag filter. This enables one queue to hold several logical streams of work while allowing different process paths, workers, or schedules to select the appropriate items. Tags are therefore a queue-item classification and selection mechanism, rather than a replacement for the queue itself. Blue Prism’s documentation describes queue-item tags and their use for filtering and retrieving queue items in the Control Room and through Work Queues actions. See [Blue Prism: Queue item tags](#) and [Blue Prism: Work queues](#).

QUESTION NO: 17 - (SIMULATION)

FILL BLANK

Examine the following recover flow taken from a Blue Prism Process:



The 'Exception Detail' data item has no initial value and has a data type of text.

The Calculation stage 'Save Exception Detail' will output the current exception detail to the 'Exception Detail' data item.

What expression would you use in the 'Save Exception Detail' stage to achieve this?

ANSWER: See the explanation for the answer

Explanation:

Use the expression:

`ExceptionDetail()`

In the **Save Exception Detail** Calculation stage, set the output to the **Exception Detail** text data item and use `ExceptionDetail()` as its expression. This built-in function returns the detail of the exception currently being handled after the Recover stage.

QUESTION NO: 18

According to Blue Prism best practice, which of the following items would it be appropriate to store in a Work Queue tag? (Choose three.)

A. Something that identifies the type of work that was performed. Example tags might be: Funds Transferred, SIM Card Requested, Account Closed, Account Suspended, Account Noted

B. Something that identifies exact datetime of a step in the process. Example of tags might include: Account Closed 20/03/2017 12:12:16, Work Completed 20/03/2017 12:12:16, Request Loaded 20/03/2017 12:12:16

C. Something that identifies the source of work. Example of tags might include: Email, Web Form, Excel, Adhoc, Database

D. Something that identifies the client record to make work queue items easier to search. Example tags might be a phone number, customer reference number or ID, customer full name. For example: +4407994446767, 00023231A, Mr John A Smith.

E. Something that identifies the type of process scenario. Example of tags might include: Invalid Product, Unrecognized customer, Gone-away Marker, Invalid Status

F. Details of an exception that has occurred, for example: The Login Screen was not found.

ANSWER: A C E

Explanation:

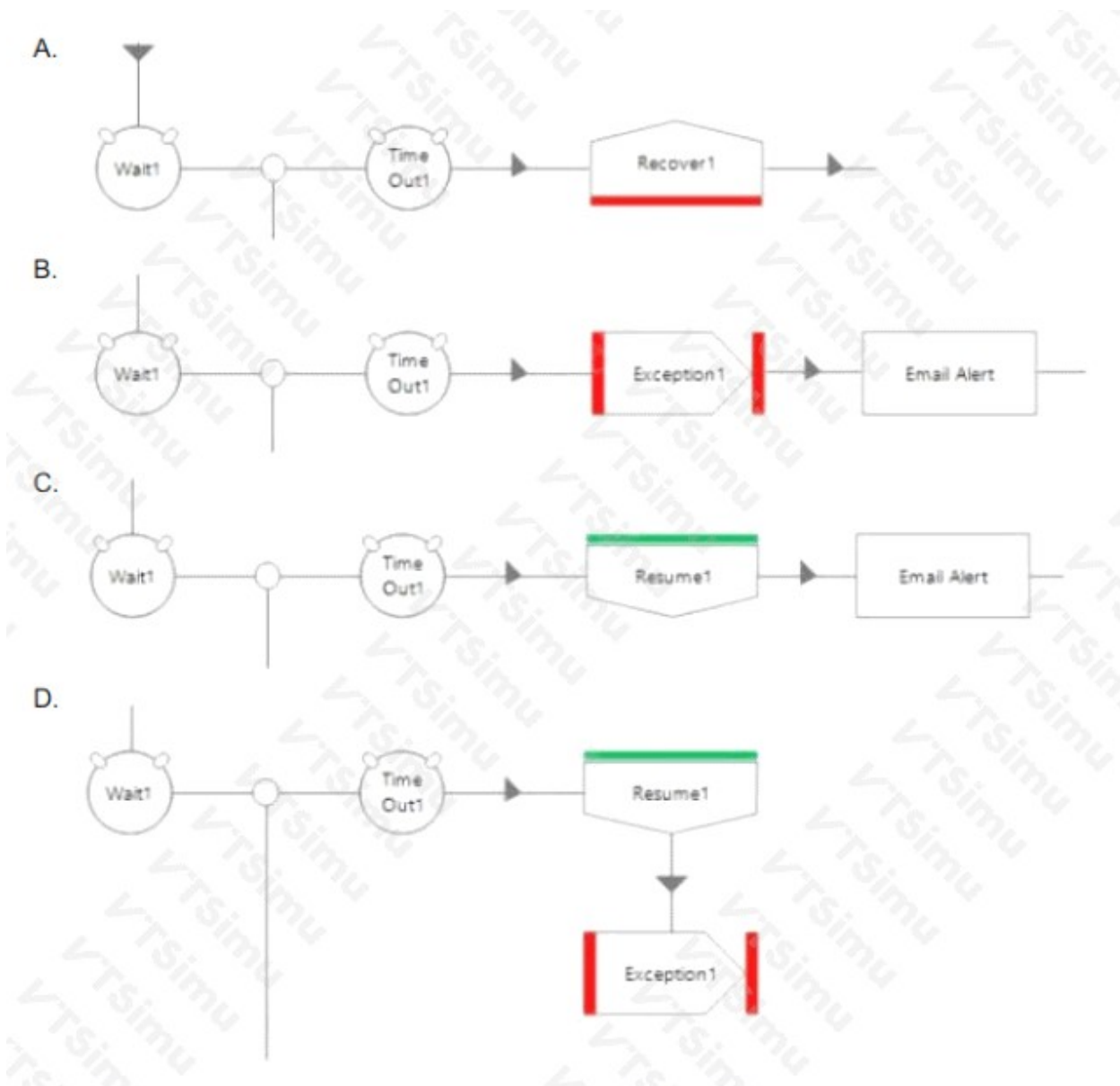
Work Queue tags are most useful when they provide consistent, reusable classifications that support operational reporting, filtering, searching, and management of queue items. “Something that identifies the type of work that was performed. Example tags might be: Funds Transferred, SIM Card Requested, Account Closed, Account Suspended, Account Noted” is appropriate because it categorizes the business activity associated with an item. This allows teams to group and review items by the service or transaction handled.

“Something that identifies the source of work. Example of tags might include: Email, Web Form, Excel, Adhoc, Database” is also appropriate because source is a stable operational classification. Capturing the intake channel in a tag helps teams monitor workload origins and identify trends in how work enters the automation.

“Something that identifies the type of process scenario. Example of tags might include: Invalid Product, Unrecognized customer, Gone-away Marker, Invalid Status” is appropriate because scenario tags distinguish meaningful business cases that may require separate monitoring or analysis. Blue Prism Work Queues support applying and using tags to organize and retrieve queue items, making concise category values valuable for queue administration. See the [Blue Prism product documentation](#) and the [Blue Prism customer portal](#) for Work Queue guidance and product resources.

QUESTION NO: 19

Which of the following flow representations are invalid and therefore cannot be recreated in a Blue Prism flow diagram? (Choose two.)



A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: B C

Explanation:

The flow representations identified as “Option B” and “Option C” are invalid because Blue Prism Process Studio and Object Studio enforce valid stage-linking rules when a diagram is created. A flow diagram is not merely a drawing: each link must represent a permissible route through executable stages, with appropriate entry into, progression through, and exit from the defined logic. Blue Prism validates the structure of the flow as stages and links are added, helping ensure that execution has an unambiguous and supported path.

Where a proposed representation would require an unsupported link arrangement or an invalid connection pattern, it cannot be reproduced simply by positioning stages to look like the illustration. The developer must instead redesign the logic using valid Blue Prism stages and links, such as decisions, calculations, actions, loops, and exception-handling constructs, while maintaining valid flow connectivity. This validation is important because the visual diagram directly defines the process or business object’s execution behavior. Blue Prism’s documentation describes Process Studio as the environment used to

create process flow diagrams and explains the use of stages and links in process design. See [Blue Prism Process Studio documentation](#) and [Blue Prism stages documentation](#).

QUESTION NO: 20

Which of the following statements about exception handling in Blue Prism are correct? (Choose two.)

- A. A Recover stage can catch an exception raised in the flow that follows it.
- B. A Resume stage ends recovery mode after an exception has been handled.
- C. A Recover stage automatically retries the stage that raised the exception.
- D. A Resume stage writes the exception detail to the session log.

ANSWER: A B

Explanation:

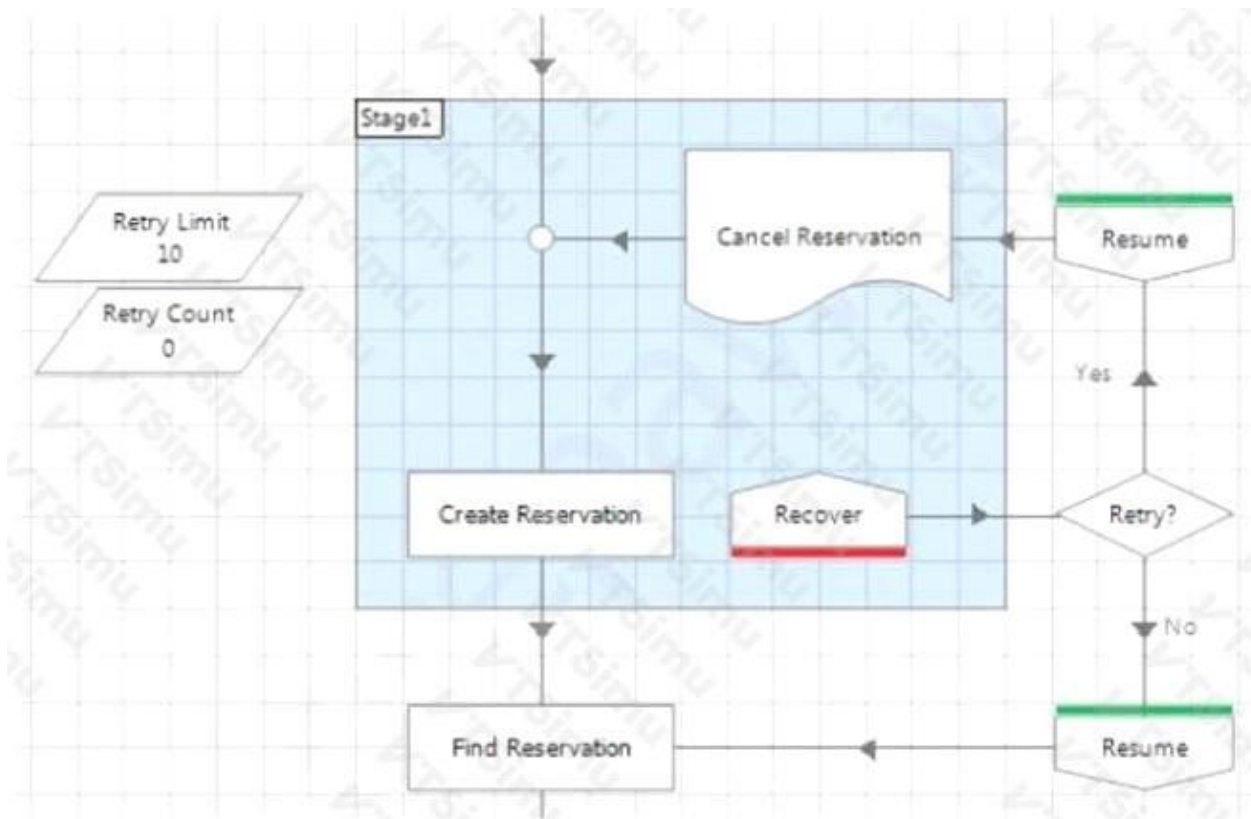
A **Recover stage** is used to catch an exception raised in the flow that follows it. Once an exception has been caught, the process or object is considered to be in recovery mode, allowing the developer to inspect the exception detail and determine the appropriate handling path.

A **Resume stage** ends recovery mode and allows normal processing to continue. A Resume stage should be used only after the exception has been handled appropriately. For example, a process may log relevant information, mark a Work Queue item as an exception, perform application cleanup, and then resume before continuing to a controlled End stage.

See the [Blue Prism documentation portal](#) and [Blue Prism University](#) for exception-handling guidance.

QUESTION NO: 21

Examine the following Blue Prism Process diagram:



What potential issues do you see with the current configuration? (Choose two.)

- A. The current configuration is a missing Recover stage on one of the routes through the diagram flow.
- B. The current configuration is a missing Resume stage on one of the routes through the diagram flow.
- C. The current configuration does not increase the Retry Count after each attempt.
- D. If an exception is thrown in the Cancel Reservation page and is allowed to bubble up to be caught by the Recover stage, the Create Reservation action won't be retried.

ANSWER: C D

Explanation:

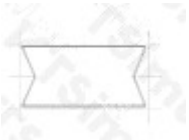
The current configuration does not increase the Retry Count after each attempt. A retry loop must maintain state that records each unsuccessful attempt and increments that state before the next attempt begins. Without increasing the counter, a decision based on the count cannot reliably enforce a retry limit. This can result in repeated attempts continuing beyond the intended maximum, particularly when the target application continues to return an exception or failure condition. The counter update needs to sit on every route that leads to another attempt, so that the retry limit reflects the total number of attempts actually made.

If an exception is thrown in the Cancel Reservation page and is allowed to bubble up to be caught by the Recover stage, the Create Reservation action won't be retried. The recovery block is arranged such that an exception from the cancellation activity reaches the same recovery handling used for the reservation attempt. Consequently, the exception path does not return to the retry route for Create Reservation. Exception handling should preserve the intended retry boundary: exceptions from the action being retried should be recovered, assessed, and routed back to the retry logic, while cleanup or cancellation exceptions require their own deliberate handling. Blue Prism's documentation describes the role of Recover and Resume stages in controlling exception handling and flow: [Blue Prism Documentation](#).

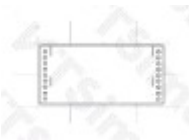
QUESTION NO: 22

Which of the following stages can be found in BOTH the Object Studio and the Process Studio ' ' Select the TWO correct responses.

A)



B)



C)



D)



E)



- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Calculation stage
- G. Decision stage

ANSWER: F G

Explanation:

The **Calculation stage** and **Decision stage** are available in both Process Studio and Object Studio. A Calculation stage evaluates an expression and assigns the resulting value to a data item or collection field. This makes it a core flow-building capability regardless of whether the developer is designing a business process or implementing the application-facing logic within an object. A Decision stage evaluates a condition and routes execution through its Yes or No path, allowing both processes and objects to control their flow according to runtime data or application state.

These two stages are therefore shared, general-purpose control and data-manipulation stages. Blue Prism distinguishes these from application-modelling stages, which are used specifically when interacting with an external application through an object, and from process-oriented stages used to organize or invoke reusable automation logic. Developers should use calculations to derive or transform values and decisions to branch the workflow cleanly, keeping business logic readable and maintainable in either studio.

See Blue Prism's [Process Studio documentation](#) and [Object Studio documentation](#) for the available stage types and their use.

QUESTION NO: 23

Which of the following can you perform in the System - > Workflow area of Blue Prism? (select all that apply)

- A. View the contents of a Work Queue
- B. Refresh a Work Queue
- C. Encrypt a Work Queue
- D. create a Work Queue

ANSWER: C D

Explanation:

In the System tab's Workflow area, developers and administrators can create a Work Queue and configure its queue-level properties. Creating a queue establishes the named, centrally managed queue that Blue Prism processes use to store and retrieve work items. This is the administrative starting point for defining how work is organized and processed across runtime resources.

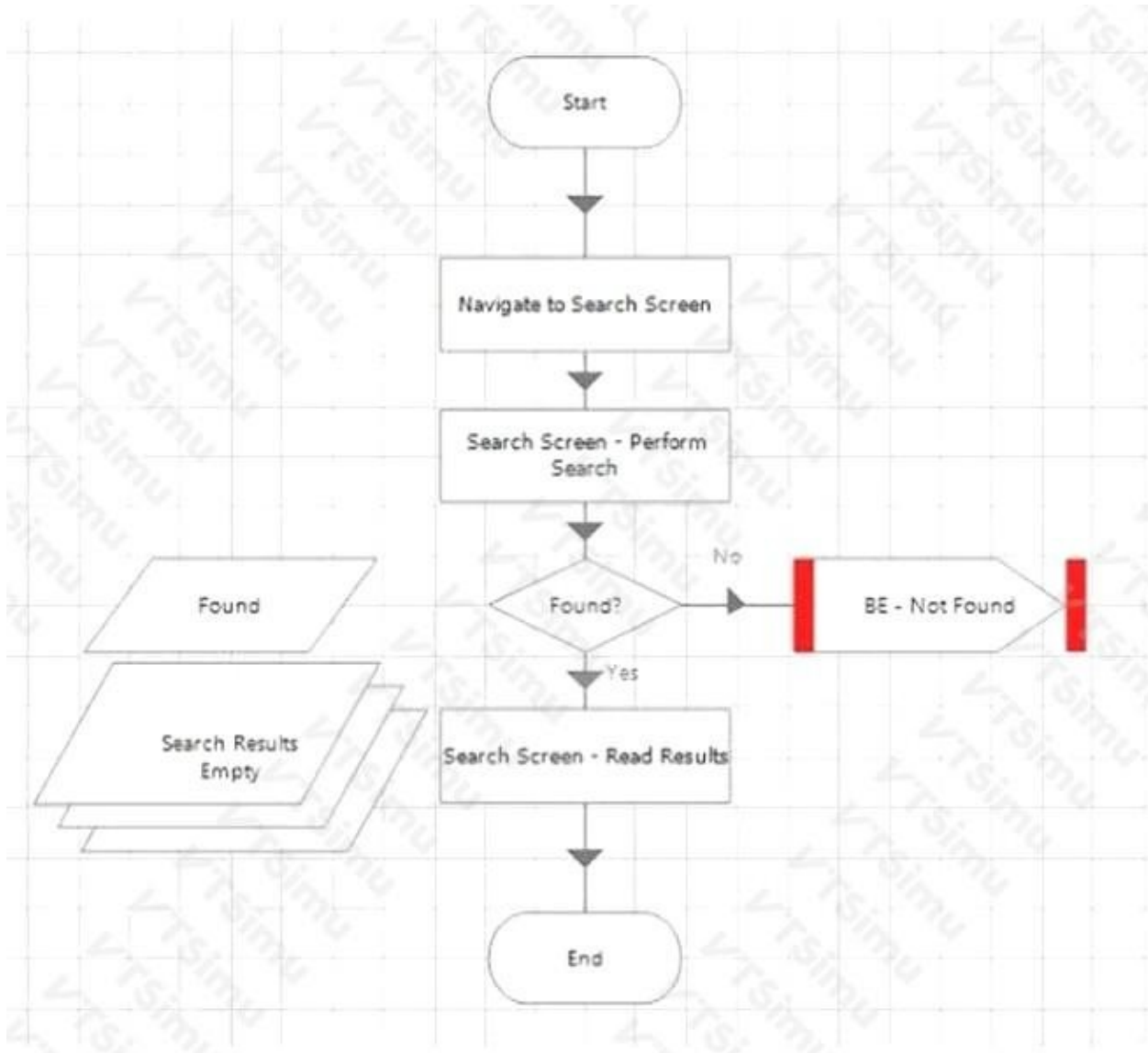
The same area supports encrypting a Work Queue through its configuration. Queue encryption protects the data held in work-queue items, which is especially important when item data may contain sensitive business information. Blue Prism

applies this setting at the queue level, so it is configured as part of the queue's administration rather than in an individual process action. A process can then interact with the queue using the standard Work Queues business object actions while the configured protection remains in effect.

Blue Prism's product documentation describes Work Queues as centrally configured resources and covers their administration and use by processes. See the [Blue Prism product documentation](#) and the [Blue Prism Digital Exchange](#) for official Blue Prism resources and guidance.

QUESTION NO: 24

Examine the following Blue Prism Process sub-page:



What is wrong with the current configuration?

- A. The current sub-page configuration is acceptable.
- B. The current sub-page configuration is missing retry exception handling. This should be included when a Process calls Business Object Actions which interface with a target application.
- C. The current sub-page configuration / stages should be moved to the Main Page for better visibility of the Process logic.
- D. The exception stage included in the current configuration should be removed and replaced with an End stage which returns the found data item as an output parameter.

ANSWER: A

Explanation:

The current sub-page configuration is acceptable. A Blue Prism process sub-page can use an Exception stage to deliberately raise an exception when the page identifies a business condition that prevents normal completion, such as data not being found. This is a valid way to communicate the outcome to the calling page: the exception propagates through the process's exception-handling flow unless it is caught by an appropriate Recover stage. It is not necessary for every sub-page that invokes a Business Object action to contain retry handling. Retry logic is a design decision based on the operation, the type of failure, and whether repeating the action is safe and likely to succeed; it should be applied only where it is meaningful. Likewise, sub-pages are an appropriate mechanism for structuring process logic into focused, reusable units, and an exception does not need to be converted into an output merely because a data item is available. The displayed use of a decision path followed by an exception outcome is therefore a sound configuration for reporting an exceptional business outcome. Blue Prism's documentation describes the role of exception stages and the Recover/Resume exception-handling model in [Exception handling](#). General process-design guidance is available through the [Blue Prism University](#).

QUESTION NO: 25

Study the following configuration

Which of the following statements is correct?

- A.** When the configuration is run in Control Room, it will become stuck in an infinite loop
- B.** When the configuration is run in Control Room, it will become stuck in an infinite loop until the 'Switch' Session Variable is changed to True in System Manager
- C.** When the configuration is run in Control Room, it will become stuck in an infinite loop until the 'Switch' Session Variable is changed to True in the Control Room
- D.** When the configuration is run in Control Room, it will fail with an internal error on the first execution of the 'Switch?' stage with the error: Decision did not result in a yes/no Answer:

ANSWER: C

Explanation:

When the configuration is run in Control Room, it will become stuck in an infinite loop until the 'Switch' Session Variable is changed to True in the Control Room. A session variable is available to the running process session and can be read each time the process reaches the decision stage. In the illustrated looping configuration, execution continues around the loop while the value evaluated by the *Switch?* decision remains false. The process does not need to be stopped and republished for the runtime value to be changed: Control Room provides the operational interface for viewing and changing the applicable session-variable value for an active session. Once *Switch* is set to True in Control Room, the next evaluation of the decision obtains that updated value. The decision then follows its true outcome, allowing the configured loop to be exited and the process to continue. This behavior is useful where an operator needs a controlled way to release a waiting or polling process without redesigning the process or altering its logic. Blue Prism's product documentation describes Control Room as the area used to monitor and manage process sessions, while system configuration and runtime resources are documented in the Blue Prism documentation portal. See [Blue Prism documentation](#) and [Blue Prism resources](#).

QUESTION NO: 26

Examine the following Process flow:

This is the Main Page of a simple process which calls the "Calculation"™ Page before completing.

The data items are configured as follows:

Value-1 is a number data item with an initial value of 100.

Value-2 is a number data item with an initial value of 4.

Result is a number data item with no initial value.

The Calculation Page Reference stage is configured as follows:

The Calculation Page is configured as follows:

The Data Items are configured as follow:

Value-1 is a Number data item with an initial value of 200

Value-2 is a Number data item with an initial value of 4

Result is a Number data item with no initial value

The Start stage is configured to pass through the Value-1 and Value-2 values as input parameters.

The "Divide" Calculation stage expression is $[Value-1] / [Value-2]$

The End stage is configured to pass back the value of the Result data item as an output parameter.

What will be contained in the Result data item on the Main Page following the execution of the Process?

- A. 25
- B. 0
- C. 50
- D. 37.5

ANSWER: A

Explanation:

25 is correct. When the Main Page calls the Calculation Page, the values configured as input parameters on the Calculation Page's Start stage are supplied by the Page Reference. The Main Page therefore passes its current *Value-1* value of 100 and *Value-2* value of 4 into the corresponding data items on the Calculation Page. These passed values are the values used during that page invocation; the Calculation Page data item's configured initial value of 200 is not used for the incoming parameter value.

The Divide Calculation stage consequently evaluates $[Value-1] / [Value-2]$ as $100 / 4$, producing 25. That value is stored in the Calculation Page's Result data item. The End stage exposes Result as an output parameter, and the Page Reference maps that output back to the Result data item on the Main Page. Once execution returns to the Main Page, its Result data item therefore contains 25.

This reflects Blue Prism's page design model: Start-stage parameters provide values to a called page, while End-stage output parameters return calculated values to the calling page. See the [Blue Prism documentation portal](#) and [Blue Prism University](#) for process and page parameter guidance.

QUESTION NO: 27

If you change the value of a 'Session Variable', when will the new value be available to a running Process?

- A. Immediately
- B. Next time a new case is taken from a Work Queue
- C. After the Digital Worker has been restarted
- D. Next time the Process starts

ANSWER: A

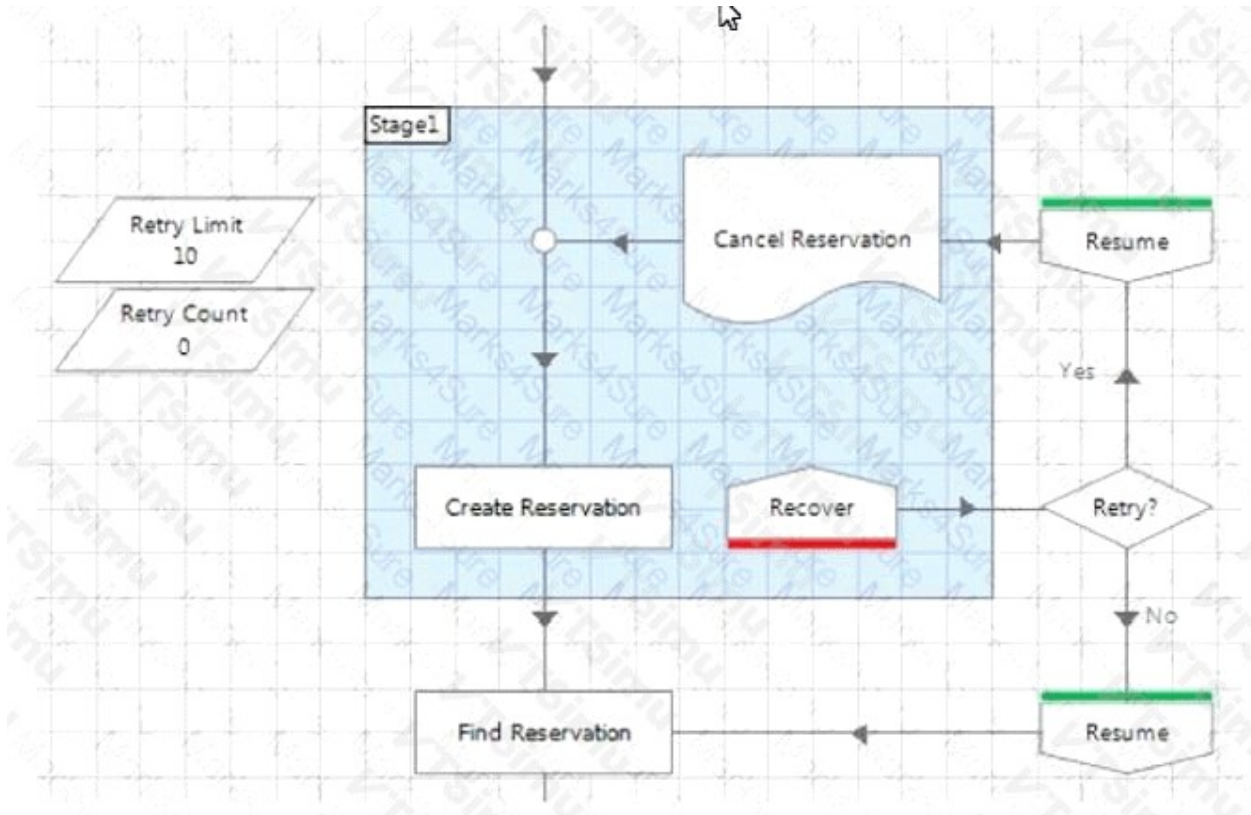
Explanation:

Immediately is correct. A Blue Prism session variable is a value held at runtime for a specific Blue Prism session and can be accessed by processes executing in that session. It is intended for information whose value may need to be shared or adjusted while the Digital Worker is operating, rather than for a value that must remain fixed until a later process run. Consequently, when the session-variable value is updated, a running process can use the updated value as soon as it next reads that session variable during its execution; it does not need to wait for a queue item boundary, a resource restart, or a

new process start. This behavior makes session variables useful for runtime control values, such as an operational flag or a temporarily changed setting that needs to take effect during active processing. Blue Prism documentation distinguishes runtime/session configuration from design-time process data and provides the administrative context for managing these values. See the [Blue Prism product documentation](#) and Blue Prism's [official community and knowledge resources](#) for current platform guidance.

QUESTION NO: 28

Study the following image showing pan: of the flow within a process:



What problems do you see with the process flow? (select 2 responses)

- A. There is a missing Recover stage on one of the routes through the flow.
- B. There is a missing Resume stage on one of the routes through the flow.
- C. The Retry Count is not being incremented for each retry.
- D. If an exception is thrown in the Cancel Reservation page and is allowed to bubble up to be caught by the Recover stage, the Create Reservation action won't be retried.

ANSWER: C D

Explanation:

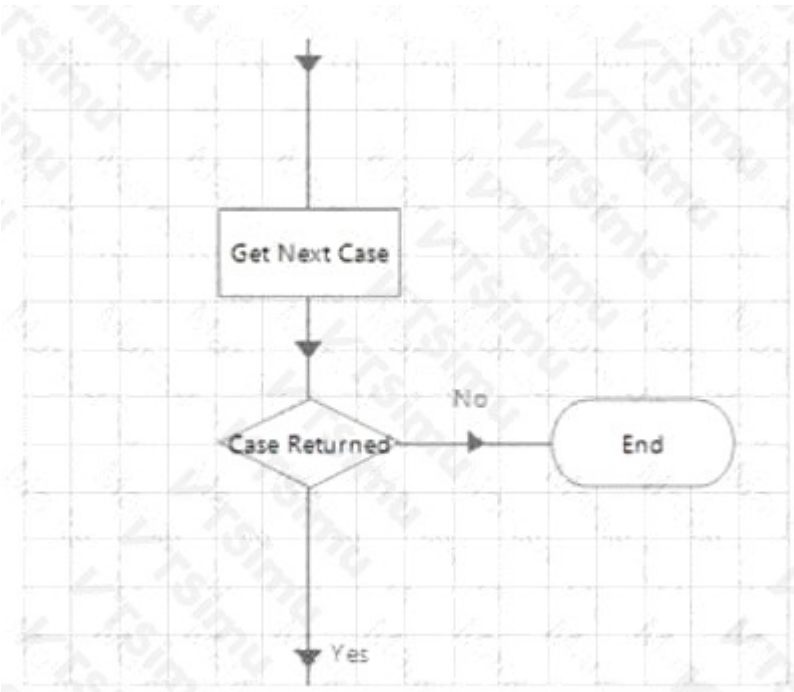
The correct issues are that *The Retry Count is not being incremented for each retry.* and that *If an exception is thrown in the Cancel Reservation page and is allowed to bubble up to be caught by the Recover stage, the Create Reservation action won't be retried.* A retry loop must update its counter on every pass before evaluating whether another attempt is permitted. Otherwise, the retry condition does not accurately represent the number of attempts already made, which can lead to repeated processing beyond the intended retry limit. The retry count should therefore be incremented as part of the route that returns to the retry decision.

Exception handling must also cover the complete unit of work that is intended to be retried. In this flow, an exception originating in Cancel Reservation can propagate to the Recover stage, but the recovery route does not return to the point that executes Create Reservation. Consequently, the recovery handling bypasses the original creation action rather than

retrying the full reservation operation. Blue Prism Recover stages establish exception handling, while Resume stages end the handling block and allow the designed recovery route to continue; the flow after recovery must explicitly return to the correct retry point. See the official [Blue Prism documentation](#) and [Blue Prism Portal](#) for product guidance.

QUESTION NO: 29

Study the process flow below:



The outputs for the Get Next Case are configured below:

Action Properties

Name:

Description:

Business Object:

Action:

Inputs Outputs Conditions

Name	Data Type	Store In
Item ID	Text	Case ID
Data	Collection	Account Details
Status	Text	
Attempts	Number	

What is the correct configuration of the Expression in the "Case Returned " decision stage?

A)



B)

Account Details <>

C)

"Case ID" <>

D)

[Case ID] <>

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: D

Explanation:

Option D is correct because the "Case Returned" decision must evaluate the result produced by the preceding "Get Next Case" action, using the output configured for that action. A decision stage expression must resolve to a Boolean value, so the expression needs to test whether the retrieval action returned a case rather than merely refer to an output value without a comparison. This allows the process to follow the case-processing path only when a valid case has been obtained and to take the alternative path when no case is available.

In Blue Prism, action outputs are exposed as data items after the action stage completes. They should be referenced in subsequent stages through the relevant process data item and evaluated in an expression appropriate to the output's configured data type. Option D applies that principle by using the returned-case output in a Boolean decision expression. This provides a reliable control point before the process attempts to use the case data and maintains the intended work-retrieval flow.

For background on building process logic and expressions, see the [Blue Prism Process Studio documentation](#) and the [Decision stage documentation](#).

QUESTION NO: 30

Which of the following options can be used to update the value of a data item exposed as a 'Session Variable', whilst the process is running?

(Select all items which apply)

A. By updating the data item value in the Variable Repository in System Manager

B. By updating the data item value in System Manager

C. Through a Calculation stage in the same process

D. Selecting the running Process in Control Room and then using the menu options to push a new value to the data item

ANSWER: C D

Explanation:

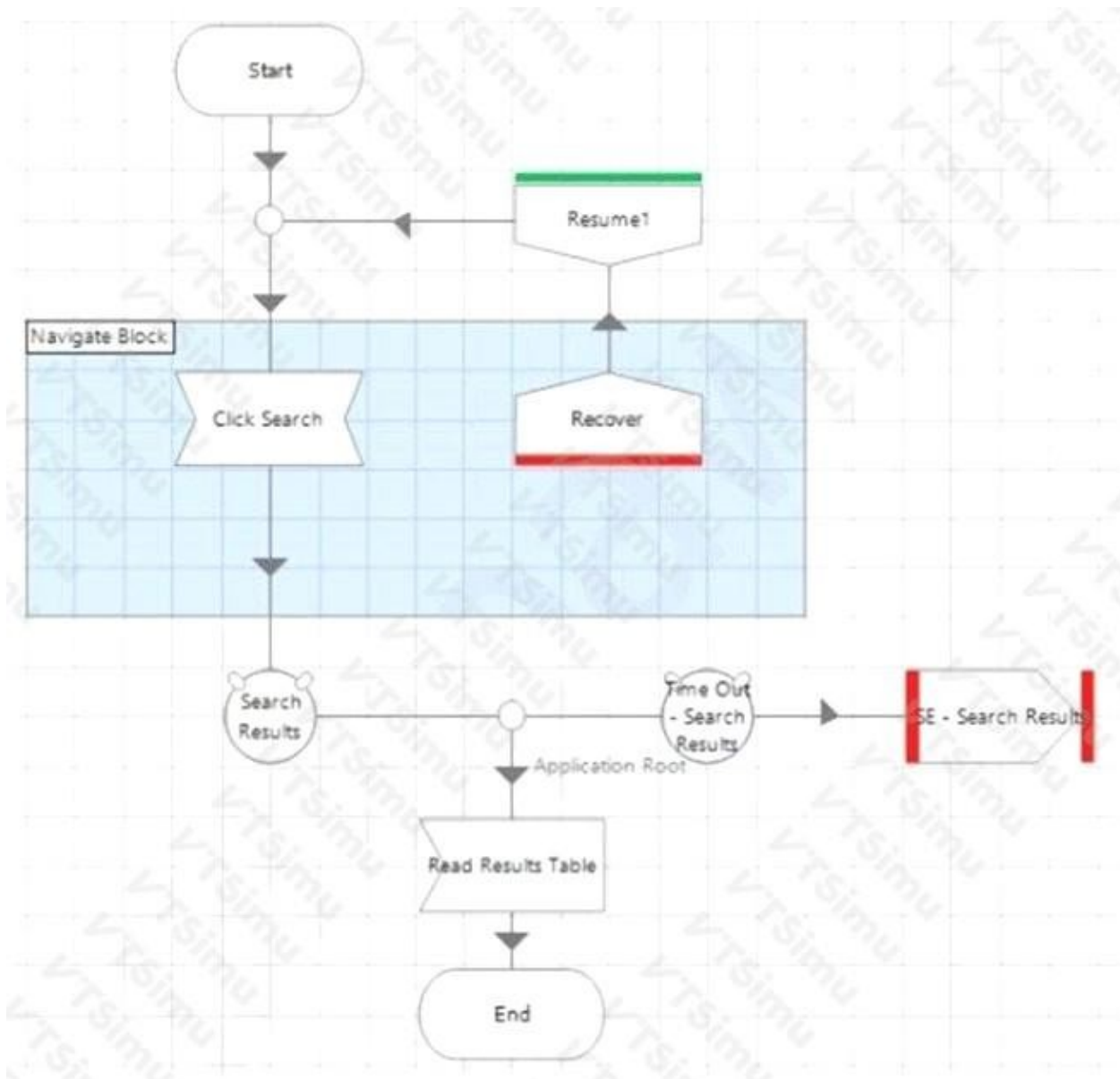
A data item configured as a session variable remains an ordinary process data item during execution, so its value can be changed by the process logic itself. “Through a Calculation stage in the same process” is therefore valid: a Calculation stage can assign a new value to the data item, and subsequent stages in that running process use the updated value. This is useful when the process must derive or revise a runtime setting from business logic, input data, or a previous result.

“Selecting the running Process in Control Room and then using the menu options to push a new value to the data item” is also valid. Exposing a data item as a session variable makes it available for authorised Control Room users to inspect and update while the session is active. This supports controlled operational intervention without stopping and restarting the process, such as changing a runtime threshold or switching a processing mode. The new value applies to the active process session and can be consumed by later process steps.

For more information, see Blue Prism’s [Control Room documentation](#) and the [Blue Prism product documentation portal](#).

QUESTION NO: 31

Examine the following Blue Prism Action diagram taken from Object Studio:



What is wrong with the current configuration?

- A. The current configuration risks creating an infinite loop.
- B. The current configuration is missing an intelligent Wait stage after the 'Read Result Table' stage, to ensure the Action logic has worked correctly.

C. There is nothing wrong with the current configuration.

D. The Exception Block should be removed from the page so that all exceptions on that page can be recovered.

ANSWER: A

Explanation:

The current configuration risks creating an infinite loop. The diagram's exception-handling route returns execution to the part of the action that can produce the same exception again, without a separate retry-control mechanism, retry counter, or termination path. If the underlying condition remains unchanged, the action can repeatedly enter the exception block, recover, and attempt the same failing work indefinitely. This can leave a runtime resource occupied, prevent the case from reaching a final outcome, and make operational diagnosis more difficult.

In Blue Prism, a Recover stage catches an exception raised within its protected block, and a Resume stage ends the handling flow and returns execution to the appropriate point after recovery. This should be designed so that recovery either resolves the cause, takes a bounded retry path, or routes the item to an explicit exception outcome. A retry design should therefore include a deterministic limit and a path that raises or records a final exception once that limit is reached. Blue Prism's exception-handling guidance describes how Recover, Resume, and exception propagation are used to manage failures safely: [Blue Prism documentation: Exception handling](#). Additional product documentation is available through the [Blue Prism Documentation portal](#).

QUESTION NO: 32

A screen contains four buttons, Find, Save, Close and New. When the buttons are spied, the attributes are identical except for the Window Text attribute.

The values for the Windows Text attribute are Find' for the Find button. 'Save' for the Save button, Close' for the Close button and New' for the New

Button.

The Dynamic Button element has been spied and has the following attributes set:

What additional configuration is required to ensure the Close button is selected by the Navigate stage?

A. Set the Mouse Button value to Close'

B. The Params button must be selected and the parameter configured to override the Window Text attribute value to 'Close'

C. It is not possible to select the Close button using the Dynamic Button element because the value is set to 'Find in the Application Modeller

D. A new element is required with the Window Text value set to "Close and the Match Type set to equal

ANSWER: B

Explanation:

The Params button must be selected and the parameter configured to override the Window Text attribute value to 'Close' because the Dynamic Button element is intended to identify one of several otherwise identical controls at run time. In Application Modeller, an element's attributes are normally fixed to the values recorded when it was spied. A dynamic attribute allows a process or business object to supply a different value when the action is called. Here, Window Text is the distinguishing attribute, so exposing that attribute as a parameter and passing `Close` causes Blue Prism to search for the button whose displayed text is Close rather than the original Find value.

The parameter is configured from the Navigate stage through the Params interface for the selected element/action. The supplied value overrides the configured dynamic attribute only for that invocation, allowing the same reusable element to operate against Find, Save, Close, or New as needed. This approach is particularly useful when controls share the same element type, parent, and other identification properties, while a single text-based attribute varies. Blue Prism's Application Modeller documentation describes the use of element attributes for identifying application controls, and its documentation library provides the relevant interface-automation guidance: [Blue Prism Application Modeller](#) and [Blue Prism Documentation](#).