DUMPSBOSS.

Designing Blue Prism Process Solutions

Blue Prism ASD01

Version Demo

Total Demo Questions: 7

Total Premium Questions: 76

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

QUESTION NO: 1

A solution is to be deployed through Development, Test and Production. The target application URL, report folder and processing timeout differ between environments, but they are not sensitive values.

Which of the following are appropriate configuration practices? (Choose two.)

A. Store the URL, report folder and timeout as environment variables.

B. Document the required configuration values for each environment.

C. Hard-code the Production URL in every business-object action.

D. Copy the configuration values into each queue item when it is loaded.

E. Store passwords in the same environment variables as the URLs.

ANSWER: A B

Explanation:

Environment variables are appropriate for configuration values that differ between environments, including URLs, folders and timeout values. The process or business object can obtain the values at run time, allowing the same released automation to be configured appropriately in each environment.

Configuration values should be centrally maintained and documented so that deployment and support teams can identify the required settings. Hard-coding environment-specific values in process logic would require changes and additional testing whenever a value changes, increasing deployment risk. Credentials remain sensitive information and should be managed through Credential Manager rather than environment variables.

See the Blue Prism documentation portal for environment and configuration management guidance.

QUESTION NO: 2

Which of the following statement combinations about Blue Prism memory management is correct?

A. A Blue Prism Process reads a Business Object into memory as required. Once the called Action is complete, the Process releases the memory for the .Net Garbage Collector to reclaim.
A Blue Prism Process reads a Sub Process into memory as required. Once the called Sub Process is complete, the Process releases the memory for the .Net Garbage Collector to reclaim.

B. A Blue Prism Process holds a Business Object in memory for the duration of its run.
A Blue Prism Process reads a Sub Process into memory as required. Once the called Sub Process is complete, the Process releases the memory for the .Net Garbage Collector to reclaim.

C. A Blue Prism Process holds a Business Object into memory as required.
Once the called Action is complete, the Process releases the memory for the .Net Garbage Collector to reclaim.

D. A Blue Prism Process holds a Sub Process in memory for the duration of its run.
A Blue Prism Process holds a Business Object in memory for the duration of its run.

ANSWER: B

Explanation:

“A Blue Prism Process holds a Business Object in memory for the duration of its run. A Blue Prism Process reads a Sub Process into memory as required. Once the called Sub Process is complete, the Process releases the memory for the .Net Garbage Collector to reclaim.” correctly describes the intended Blue Prism runtime behaviour. When a process first uses an action from a business object, Blue Prism loads the object and retains it for the remainder of that process execution. This allows later actions from the same object to be invoked without repeatedly loading the object, and it preserves the object instance and its state while the process is running. A subprocess has a different lifecycle: it is loaded when the calling process reaches the subprocess call, and its runtime resources can become eligible for garbage collection once that subprocess has returned and no references remain. This distinction matters in solution design, particularly where a long-

running process invokes many subprocesses or uses objects that maintain application sessions or cached state. Designers should therefore structure process and object calls with these lifecycles in mind, while allowing the .NET garbage collector—not the process author—to determine the precise timing of physical memory reclamation. Blue Prism product documentation is available at Blue Prism Documentation, and the official learning portal is available at Blue Prism University.

QUESTION NO: 3

A solution is deployed to Development, Test and Production environments. The URL of a target application differs in each environment, but the business object actions and process logic are otherwise identical.

What is the most appropriate way to manage the URL?

A. Store the URL in an Environment Variable.

B. Store the URL in a global data item in every process.

C. Hard-code the URL in each Navigate stage that uses the application.

D. Store the URL as the reference of a Work Queue item.

ANSWER: A

Explanation:

The URL should be held in an Environment Variable. Environment Variables enable values that differ between environments to be configured without changing the process or object design. The business object can obtain the configured value when it launches or attaches to the application.

Storing the URL directly in a calculation stage or object action would require a release whenever the value changes and increases the risk that a Development value is promoted to Production. Environment Variables support controlled, environment-specific configuration. See the Blue Prism documentation portal for environment and release-management information.

QUESTION NO: 4

Which of the following statements about using Work Queue designs to split a business process into a multi part robotic solution is correct?

1. Using multiple Work Queues and Processes for the different stages of the business process is a valid design option to split a business process into a multi part robotic solution.

2. Using a single Work Queue and deferring cases for future processing is a valid design option to split a business process into a multi part robotic solution.

3. Using an item's status to control when to work it is a valid design option to split a business process into a multi part robotic solution.

4. You cannot split a business process into a multi part robotic solution.

A. 1 and 2 Only

B. 4 only

C. 1 and 3

D. 1, 2 and 3

ANSWER: A

Explanation:

1 and 2 Only is correct. Blue Prism Work Queues provide a durable mechanism for separating a larger business workflow into discrete, independently run robotic stages. One appropriate design is to use separate queues and processes for each stage: a process completing one stage can place the required data onto the queue used by the next stage. This makes the hand-off explicit, supports separate scheduling and resource allocation, and gives each stage its own operational monitoring and exception-handling boundary.

A single queue can also support a multi-part solution when an item must wait until a later time, event, or eligibility point before further work occurs. In that design, the process defers the item and assigns an appropriate deferred-until time. Blue Prism keeps the item available for later processing rather than treating it as complete, allowing robots to retrieve it once it becomes due. This is useful where the same overall work item has a controlled pause between stages, such as waiting for an external response or a scheduled processing window. Blue Prism's queue-management guidance describes queue items, their lifecycle, and deferred processing; see the Blue Prism documentation portal and Blue Prism guides and documentation resources.

QUESTION NO: 5

Several processes need to calculate a settlement date using the same calendar rules. The calculation does not interact with any external application, and changes to the rules must be applied consistently to every process.

Which of the following is the best design option?

A. Create a reusable subprocess for the settlement-date calculation.

B. Place the calculation in a separate business object for each process.

C. Duplicate the calculation in each process so that they remain independent.

D. Store the calculated date in a shared spreadsheet for all processes to use.

ANSWER: A

Explanation:

The common calculation should be implemented as a reusable subprocess with defined inputs and outputs. Each process can call the subprocess with the required date and receive the calculated settlement date. This centralises the business rule, reduces duplicated logic, and ensures that a future rule change can be made and tested in one place.

A business object would be more appropriate if the logic were encapsulating interaction with an external application or service. Duplicating the calculation in every process creates avoidable maintenance effort and risks inconsistent implementation of the business rules. See the Blue Prism documentation on processes.

QUESTION NO: 6

A process is required for a client in the banking sector that involves using an application to transfer funds between accounts. A strong security model is in place to prevent any malicious activity but the client is nervous about the risk of external problems, like a power cut leaving a case in an incomplete state or a fault in the source data causing a case to be duplicated or an excessively large transfer to be made. What should be included in the solution design? (Choose three.)

A. A different queue for each major processing step.

B. A different process for each major processing step.

C. A single queue that is cleared of all worked items at the start of each day.

D. Rules to limit transaction values.

E. A key value that will uniquely identify queue items.

F. A requirement that the solution is never run on more than one machine.

G. An exception handling procedure to track manual referrals.

ANSWER: B D E

Explanation:

A different process for each major processing step supports a resilient, restartable design. Dividing a long end-to-end transfer flow into defined process boundaries makes it easier to establish clear completion points, isolate failures, and resume or reconcile outstanding work after an unexpected interruption such as a power outage. This is particularly important where a transfer may have been submitted to an external application but the automation has not yet recorded its final outcome.

Rules to limit transaction values provide an important business control for financial processing. The solution can validate an amount against agreed thresholds before submitting it, allowing transactions outside the permitted range to be referred for appropriate review rather than being processed automatically.

A key value that will uniquely identify queue items protects the integrity of the workload. Blue Prism work queues support a unique reference for an item, enabling the solution to identify duplicate source records and avoid creating or processing the same instruction more than once. Together, these measures provide recoverability, controlled processing, and duplicate prevention. Blue Prism documentation on platform features and work-queue capabilities is available at Blue Prism Documentation and Blue Prism Resources.

QUESTION NO: 7

Consider the following steps for a theoretical manual process.

- Check in input folder any new files.
- If there are no files check again later as files can arrive anytime, and there is no limit to the number of files that may come.
- Open the next available file.
- Take the first case.
- Start System X and find the case details.
- If the case can't be found, move to the next one.
- After finding the case in System X, fetch additional case details from System Y.
- Again if the case can't be found, move to the next one.
- Analyse all the data to see if System Z should be updated.
- If the data does not meet the requirements, add notes indicating this to System X and Y and move to the next case.
- If the data does meet the requirements, update the case in System Z.
- Add notes to System X and Y and move to the next case.
- At the end of the file, go back and look for another.
- Stop checking for new files at 16:00 and finish any remaining cases.
- When all work is complete create a report of the day's exception cases.
- Close down Systems X, Y and Z.

If it is possible that there are long intervals between files arriving, what are the alternatives the process should take to control System X, System Y and System Z? (Choose two.)

A. Nothing because they will not have been started if no files have arrived yet.

B. Nothing, just leave them logged in until more work arrives.

C. Minimize them to keep the desktop clear.

D. Close Y and Z down but keep X open because it is the first application needed to work a case.

E. Log out while waiting and log back in when a file arrives.

F. Stop them from timing out by keeping them active while waiting for files.

ANSWER: B E

Explanation:

For long, unpredictable periods without incoming files, the design can legitimately use either of two defined session-management strategies: “Nothing, just leave them logged in until more work arrives.” or “Log out while waiting and log back in when a file arrives.” The appropriate choice depends on the target applications’ session-timeout behavior, licensing and capacity constraints, security policy, and the cost and reliability of establishing a new session. Retaining authenticated sessions can reduce repeated start-up and sign-in activity, provided that the applications tolerate idle sessions and this complies with operational policy. Alternatively, explicitly logging out during an extended wait releases application sessions and reduces the risks associated with unattended authenticated access; the process must then include robust, repeatable login steps before it resumes case handling. In either approach, the decision should be intentional and implemented as part of the process’s normal application lifecycle, including orderly closure at the end of processing. Blue Prism designs should account for the availability and state of runtime resources and applications throughout execution; see the Blue Prism documentation portal and Blue Prism’s implementation guides for operational design guidance.