DUMPSBOSS.

Automation Anywhere Certified Advanced RPA Professional (V11)

Automation Anywhere Advanced-RPA-Professional

Version Demo

Total Demo Questions: 24

Total Premium Questions: 244

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Automation Anywhere Enterprise Platform	93
Topic 2, Bot Development	109
Topic 3, MetaBot Development	19
Topic 4, Error Handling	12
Topic 5, Bot Deployment	11
Total	244

QUESTION NO: 1

Which mode of Information Capture is supported by Smart Recorder?

A. Command Mode

B. Text Mode

C. Window mode

D. Object mode

ANSWER: D

Explanation:

Object mode is supported by Automation Anywhere's Smart Recorder because Smart Recorder is designed to identify and capture user-interface elements as application objects. During recording, it recognizes controls such as buttons, text fields, lists, menus, check boxes, and similar elements, then stores the interaction using properties that identify the selected object. This object-based approach enables a bot to interact with the intended control rather than relying only on a fixed screen position. As a result, the recorded automation can use the application's UI structure and object attributes when it runs.

In Automation Anywhere Enterprise version 11, Smart Recorder is the recorder intended for working with Windows and web application objects, including situations where the bot must recognize and act on individual controls. The Smart Recorder documentation and the Enterprise version 11 documentation set describe this object-oriented recording capability and its use for capturing application interactions. See the official Smart Recorder documentation and the Enterprise v11 Bot Creator documentation.

QUESTION NO: 2

A developer is designing a bot that updates records in a database. The bot must establish a database session before processing and release the session after all database operations are complete.

Which two Database command actions are required to manage the database session? (Choose two.)

A. Connect

B. Disconnect

C. Execute SQL

D. Run Stored Procedure

ANSWER: A B

Explanation:

Connect and **Disconnect** are correct. The Connect action establishes the database connection that is required before the bot can execute SQL statements or call stored procedures. The bot uses the active connection while performing its database work.

The Disconnect action closes the connection after the required operations have completed. Closing connections appropriately is important because it releases database and network resources and prevents the bot from retaining unnecessary sessions. Execute SQL and Run Stored Procedure perform database operations, but they do not establish or close the database session. See the Automation Anywhere Enterprise v11 Database command documentation.

QUESTION NO: 3

What is the pre-defined system role associated with Bot Insight dashboard?

A. 'AAE_Bot Insight Admin' and 'AAE_Bot Insight Consumer'

B. 'AAE_Admin' and 'AAE_Bot Insight Consumer'

C. 'AAE_Bot Insight Admin', AAE_Bot Insight Expert and 'AAE_Bot Insight Consumer'

D. 'AAE_Bot Insight Expert' and 'AAE_Bot Insight Consumer'

ANSWER: D

Explanation:

'AAE_Bot Insight Expert' and 'AAE_Bot Insight Consumer' are the predefined roles associated with using Bot Insight dashboards in Automation Anywhere Enterprise v11. The Bot Insight Expert role is intended for users who design and manage dashboard content, including creating and configuring dashboards, widgets, and the associated data visualizations. The Bot Insight Consumer role is intended for business users and other dashboard viewers who need to access and consume the published dashboard information for operational monitoring and decision-making. This separation supports role-based access control by allowing dashboard authors to build analytics while granting appropriate view access to consumers. Automation Anywhere documents predefined roles in the Control Room as part of its user and role management model, and Bot Insight documentation describes dashboards as the interface for presenting bot and business analytics. See the Automation Anywhere Enterprise v11 documentation on managing users and roles and the Bot Insight overview.

QUESTION NO: 4

A user has made a MetaBot that will populate data into a web form after reading values from a database. Variables have been made in the MetaBot which will accept values from their counterparts in the calling Task, however the user

is not seeing these variables on calling the MetatBOt in the Taskbot.

Why is this occurring?

A. The MetaBot variables need to be declared as type Output

B. The MetaBot variables need to be declared as type Visible

C. The MetaBot variables need to be declared as type Input

D. The MetaBot variables need to be declared as type Editable

ANSWER: C

Explanation:

The MetaBot variables need to be declared as type Input because an Input variable is the MetaBot interface mechanism for receiving a value supplied by its calling TaskBot. When a MetaBot is invoked from a TaskBot, Automation Anywhere displays the exposed input parameters in the MetaBot command configuration, allowing the developer to map each parameter to a TaskBot variable, a literal value, or other supported source. The value is then available to the MetaBot logic and can be used to populate the target web form.

Creating a variable within the MetaBot alone does not automatically expose it to the calling TaskBot. The variable must be configured with the appropriate input direction so that it becomes part of the MetaBot's callable contract. This separation lets a MetaBot remain reusable: the TaskBot can retrieve database data and pass different records into the same MetaBot without modifying its internal implementation. Automation Anywhere's documentation describes MetaBots as reusable automation components and identifies input/output variables as the means for exchanging data with a calling TaskBot. See Automation Anywhere Documentation and Automation Anywhere University.

QUESTION NO: 5

A user is planning to use the Import Bot files tool in the Automation Anywhere Enterprise Control room. When the user browses to select the bot files to import they find the folder empty.

What is the cause of this issue?

A. Bot files must be saved to the My Tasks\Export folder before running the Export task on the source Control Room

B. The Export has not been performed with the necessary steps.

C. Bot files can only be imported from the My Tasks\Import folder

D. The Import destination host on the source Control Room needs to be defined

ANSWER: B

Explanation:

The Export has not been performed with the necessary steps. is correct because the Control Room import process relies on an exported Bot package being created first from the source Control Room. Exporting prepares the selected Bot files and their required related assets in a transferable package, which can then be made available for selection during import into the target Control Room. If the expected location is empty while browsing in the Import Bot files workflow, the required export artifact has not been generated, downloaded, or placed where the importing user is browsing. The user should return to the source environment, use the Bot export workflow for the required files, complete the export, and ensure the resulting package is available to the machine and user session performing the import. This export-before-import sequence helps preserve the Bot dependencies and metadata needed for a successful migration between Control Rooms. Automation Anywhere documents Bot migration as separate export and import operations; review the Enterprise v11 Control Room documentation for the relevant procedures: Export Bot files and Import Bot files.

QUESTION NO: 6

Which two are the available recorders in Automation Anywhere?

A. File Recorder

B. Command Recorder

C. Web Recorder

D. Smart Recorder

ANSWER: C D

Explanation:

In Automation Anywhere Enterprise v11, **Web Recorder** and **Smart Recorder** are available recorder types for capturing user actions and turning them into automated bot tasks. Web Recorder is intended for recording interactions within supported web browsers and web applications. It identifies web elements and generates actions that the bot can replay, making it useful for browser-based processes such as entering data, clicking controls, and navigating pages.

Smart Recorder is designed for recording interactions with desktop applications as well as web applications. It uses object recognition to identify application controls and supports more resilient automation across different user-interface technologies. During recording, Automation Anywhere captures the selected objects and related actions, which Bot Creators can subsequently review and refine in the task editor. These recorder capabilities help accelerate bot development while still allowing the recorded commands to be edited to suit the process requirements. Automation Anywhere's Enterprise v11 documentation describes the recorder features and their use in bot creation: Automation Anywhere Enterprise v11 recorder documentation.

QUESTION NO: 7

A developer is building a TaskBot that must process every file in an input folder and repeat a validation action five times for each file.

Which two Loop command options can support this design? (Choose two.)

A. Each File in a Folder

B. Number of Times

C. Each Window in an Application

D. Each Cell in a PDF

ANSWER: A B

Explanation:

Each File in a Folder and **Number of Times** are correct. Each File in a Folder allows the bot to iterate through the files found in a specified directory and perform the enclosed actions for every file. This is appropriate for processing a variable number of incoming files.

Number of Times allows the bot to repeat a group of commands a specified number of times, such as five validation attempts or five repeated checks. Combining these loop types allows the developer to process each file and apply the required repeated action within that file-processing workflow. See the Automation Anywhere Enterprise v11 Loop command documentation.

QUESTION NO: 8

It is possible to run/execute a task by pressing a pre-assigned Hot Key combination which has been associated with the task beforehand (True/False)?

A. True

B. False

ANSWER: A

Explanation:

True is correct. In Automation Anywhere Enterprise v11, a task can be configured with an assigned keyboard shortcut, commonly called a hot key. After the hot key is associated with the task, a user can start that task by pressing the configured key combination, rather than locating and selecting the task manually from the Automation Anywhere client. This capability is useful when a frequently used automation must be launched quickly from the user's workstation.

The assigned combination must be available and should be chosen carefully so that it does not conflict with an operating-system shortcut or another application's shortcut. The task must also be available to the logged-in Automation Anywhere user, and the Automation Anywhere client must be running in the applicable environment for the shortcut to invoke the automation. Hot-key execution is a task-launching convenience feature; it does not remove the normal requirements for task permissions, bot availability, or any required credentials and runtime configuration.

Automation Anywhere's Enterprise v11 documentation describes client-side task and bot execution capabilities in its hot-key documentation and the broader Bot Runner task-execution documentation.

QUESTION NO: 9

What are two valid Error Handling options? (Choose two.)

A. Stop

B. Continue

C. Continue until next error

D. Stop after next error

ANSWER: A B

Explanation:

In Automation Anywhere Enterprise v11, the Error Handling command lets a bot define what should happen when a command encounters a runtime error. **Stop** is a valid setting because it ends execution when the configured error condition is encountered. This behavior is appropriate when continuing could produce incomplete, unreliable, or potentially unsafe business results, such as when a required application is unavailable or a critical data update fails.

Continue is also a valid setting because it instructs the bot to proceed with subsequent commands after the error. This can be useful when an error is noncritical, when later steps can still complete independently, or when the bot has been designed to record the issue for later review. Error-handling behavior should be selected according to the business impact of the failed operation and the bot's recovery design.

The Enterprise v11 command documentation describes Error Handling as the mechanism for controlling bot execution after errors occur. See the Automation Anywhere Enterprise v11 Error Handling command documentation and the Automation Anywhere documentation portal for product documentation and guidance.

QUESTION NO: 10

PDF Integration- Extract Text command allows output in what two different text-types?

A. Plain and Comma Separated

B. Un-formatted and Formatted

C. Plain and Structured

D. Structured and Tabular

ANSWER: C

Explanation:

Plain and Structured is correct. In Automation Anywhere Enterprise v11, the PDF Integration package's Extract Text action supports two output text types: plain text and structured text. Plain text extracts the PDF's text content as a continuous, unstructured text result. This is useful when the bot only needs searchable or readable content and does not need to retain layout-related organization.

Structured text is intended for PDFs where the organization of extracted content matters. It preserves information in a structured representation so that the bot can work more effectively with content that has meaningful document layout, such as rows, columns, or other grouped elements. Selecting the appropriate type lets a bot author align extraction output with the next automation step, whether that step is simple text processing or processing content whose structure must be retained.

Automation Anywhere documents PDF commands and their extraction capabilities in its Enterprise v11 documentation. See the PDF Integration command documentation and the guidance for using PDF Integration.

QUESTION NO: 11

An administrator is tasked to consume work items from a given queue in the Control Room instance in an existing connection. In the Variable Manager panel, what must the administrator first add in the Bot?

A. Queue Variables

B. Queue Category

C. Work Item

D. Bot Dependency

ANSWER: B

Explanation:

Queue Category is correct because a bot must first define its queue reference in Variable Manager before it can consume work items from a Control Room queue. Selecting the Queue category allows the bot developer to create a queue variable and associate it with the required Control Room connection and queue. This establishes the queue context that Workload Management actions use when retrieving work items for processing.

Once the queue variable is configured, the bot can use the applicable Workload Management command to obtain the next available item, process its data, and update the item's status as processing progresses. The queue definition is intentionally maintained as a bot variable so the automation has a controlled, reusable reference to the intended Control Room workload source rather than relying on an unconfigured queue name at runtime. This setup supports reliable workload distribution across bots and enables Control Room to manage item availability and processing state centrally. Automation Anywhere documents workload queues as the Control Room mechanism for distributing work items to bots; see the Automation Anywhere Enterprise v11 Workload Management documentation and the Automation Anywhere documentation portal.

QUESTION NO: 12

PDF Integration- Extract Text command allows output in two different text-types, which are

_____.

A. Un-formatted

B. Tabular

C. Comma Separated

D. Plain

E. Structured

ANSWER: D E

Explanation:

The PDF Integration package's Extract Text command supports the two text-output types **Plain** and **Structured**. Plain text output extracts the readable character content from the selected pages as a continuous text result, making it appropriate when a bot needs to search for phrases, validate a value, or write simple extracted content to a variable or file. Structured text output retains layout-related information from the PDF extraction, such as the positioning and organization of text elements. This is useful when the bot must work with content whose visual arrangement matters, including documents that contain aligned fields, columns, or table-like layouts.

The selected text type controls the form of the extracted result; it is not merely a file-format choice. A bot developer should select Plain when downstream automation only requires text content and select Structured when preserving document organization will make later parsing or interpretation more reliable. This behavior is documented as part of Automation Anywhere's PDF package command capabilities for Enterprise version 11. See the Automation Anywhere Enterprise v11 PDF package documentation and the Automation Anywhere documentation portal.

QUESTION NO: 13

A user scheduled some tasks on the local client and left for the day. The next day, the user finds that none of the tasks were executed. The event log shows that the machine did not shut down or restart.

Which of these are possible reasons that the command did not execute? (Choose two.)

A. The Auto Login feature was not set up on the machine

B. The Automation Anywhere Enterprise Scheduler Service got shut down

C. The Automation Anywhere Enterprise Client Service got shut down

D. The Visual SVN Server service got shut down

ANSWER: A B

Explanation:

The Auto Login feature was not set up on the machine is a possible cause because locally scheduled Automation Anywhere tasks may need an active Windows session at their scheduled time. Auto Login is designed to make the required user session available automatically when an unattended local execution is due. If the user signed out or the workstation was locked in a state that prevents the scheduled task from obtaining its required desktop session, the command cannot begin even though the computer itself remained powered on.

The Automation Anywhere Enterprise Scheduler Service got shut down is also a direct cause. The Enterprise Scheduler Service is the Automation Anywhere component responsible for monitoring schedule definitions and triggering commands when their configured execution time arrives. If that service is stopped, scheduled commands are not dispatched, regardless of whether the Windows event log shows a shutdown or restart. Checking that the service is running, configured with an appropriate startup type, and able to access the local Enterprise Client installation is an important troubleshooting step for missed schedules. Automation Anywhere scheduling documentation describes using schedules to run tasks at specified times, while Windows service guidance explains that a stopped service does not perform its background processing until it is started again. See Automation Anywhere Enterprise v11 scheduling documentation and Microsoft Windows Services documentation.

QUESTION NO: 14

A developer configures Error Handling so that a bot can capture technical details when a command fails. The developer wants to store the error information in variables for use in an error email and a log file.

Which two system variables provide error details? (Choose two.)

A. ErrorNumber

B. ErrorDescription

C. ArrayRows

D. ArrayColumns

ANSWER: A B

Explanation:

ErrorNumber and **ErrorDescription** are correct because they provide information about the error encountered during bot execution. ErrorNumber identifies the error, while ErrorDescription provides the associated message or descriptive information that can help a developer or support team investigate the failure.

These system variables are useful in an Error Handling block when a bot must record diagnostic information, notify support personnel, or determine an appropriate recovery path. ArrayRows and ArrayColumns are system variables associated with array dimensions and do not provide error information. See the Automation Anywhere Enterprise v11 Error Handling command documentation.

QUESTION NO: 15

Which of the following actions can be performed using the PDF Integration commands?

A. Extraction of Text from PDF

B. Extraction of Form-Fields from PDF

C. Conversion of PDF into Image

D. Conversion of Text into PDF

ANSWER: A B C

Explanation:

Extraction of Text from PDF, Extraction of Form-Fields from PDF, and Conversion of PDF into Image are supported uses of the Automation Anywhere Enterprise v11 PDF Integration command set. Text extraction enables a bot to retrieve textual content from PDF documents for downstream processing, such as validation, searching, data parsing, or entering information into another application. Form-field extraction enables a bot to access values held in interactive PDF form fields, which is useful when processing completed business forms without manually opening each file. PDF-to-image conversion allows PDF pages to be rendered as image files, supporting workflows that require visual review, image-based archival, or use with other image-processing capabilities. These actions reflect the package's purpose of reading and transforming information already contained in PDF files as part of an automated workflow. Automation Anywhere documents PDF Integration among its Enterprise v11 command capabilities; see the PDF Integration documentation and the Enterprise v11 command reference.

QUESTION NO: 16

A developer creates a script from the Workbench with the debugging mode set to 'ON' but is not seeing any Images in the Visualize section. What needs to be set to view the images?

A. Images can be captured manually via any Image Capture utility

B. Setting Breakpoints

C. Set Snap point

D. Toggle Breakpoint

ANSWER: C

Explanation:

Set Snap point is correct because Debug mode alone enables diagnostic execution, but it does not automatically create the visual snapshots displayed in the Visualize section. In Automation Anywhere Enterprise v11 Workbench, a SnapPoint is placed at the relevant step in a task so that, when the bot runs in Debug mode, Automation Anywhere captures the screen state at that point. The captured image is then available in Visualize, allowing the developer to inspect what the bot encountered during execution and troubleshoot the task with contextual evidence.

SnapPoints are particularly useful around steps that interact with applications, controls, windows, or data whose runtime state may differ from the developer's expectation. The bot must execute the step containing the SnapPoint during the debug run for a corresponding visual capture to be produced. This behavior makes SnapPoints the required configuration for generating Visualize images, while Debug mode supplies the execution context in which those captures are collected. Automation Anywhere documentation and learning resources are available through the Automation Anywhere Documentation portal and the Automation Anywhere University.

QUESTION NO: 17

Which two reports are generated by the Report Designer? (Choose two.)

A. Workflow Failure

B. Task Run

C. Task Failure

D. Workflow Run

ANSWER: B D

Explanation:

In Automation Anywhere Enterprise version 11, Report Designer supports reports that provide execution-focused information at both the individual task level and the broader workflow level. **Task Run** is generated to present task-execution details, enabling users to review runs associated with deployed or scheduled automation tasks. This supports operational monitoring by making task-run information available in a report format.

Workflow Run is also generated by Report Designer and provides reporting on workflow execution. A workflow can coordinate automation activity across tasks, so workflow-run reporting gives users visibility into the execution history and status of that higher-level process. Together, these report types allow administrators and business users to examine automation performance from the task perspective as well as the workflow perspective.

Automation Anywhere documents reporting and Control Room monitoring capabilities for Enterprise version 11 in its product documentation. See the Automation Anywhere Enterprise v11 reports documentation and the Enterprise v11 Control Room documentation.

QUESTION NO: 18

An administrator captures the value of $Error Line Number$ variable in a log file and finds that the value of $Error Line Number$ variable is 4. What does this mean?

A. There is an error on line number 4 of the script

B. There is an error in one of the first three lines of the code

C. All of the first four lines of the code contain an error

D. Lines 1 through 4 are correct, but there is an error in the script after this point

ANSWER: A

Explanation:

`$Error Line Number$` is an Automation Anywhere system variable that records the line number in the bot script at which the most recent error occurred. Therefore, when the logged value is `4`, it identifies line 4 as the location where Automation Anywhere detected the error. This value is particularly useful when error handling or logging is configured, because an administrator can use it with other error-related system variables, such as the error description, to quickly locate the failing command in Bot Creator and investigate the relevant input, application state, credentials, or command configuration. The line number refers to the bot script's numbered command line, not a count of how many errors occurred and not a statement about whether preceding or following lines are valid. Automation Anywhere documents system variables as built-in variables that provide execution and runtime information for bots. For the Enterprise v11 documentation, see the Automation Anywhere system variables reference. Additional guidance on using error handling in bots is available in the Automation Anywhere error-handling documentation.

QUESTION NO: 19

A developer is creating a TaskBot that must repeat the same set of actions for every file and every subfolder located in a specified directory.

Which two Loop command options can be used? (Choose two.)

A. Each File in a Folder

B. Each Folder in a Folder

C. Each Window in an Application

D. Each Credential in a Locker

ANSWER: A B

Explanation:

Each File in a Folder is a valid Loop command option because it iterates through the files found in a selected directory. The TaskBot can use the current file information during each iteration to open, validate, move, rename, or process the file.

Each Folder in a Folder is also valid because it iterates through the subfolders contained in a selected parent folder. This is useful where an automation must process directory-based batches or navigate an organized folder structure.

These loop types avoid hard-coding individual paths and allow a TaskBot to adapt to changing file volumes. Automation Anywhere documents loop options in the Enterprise v11 Loop command documentation.

QUESTION NO: 20

A user is developing a TaskBot in which a business process will use a MetaBot to perform data entry via a web application. Information extracted from form fields in a set of PDFs is not being entered into the third-party site.

What is causing the problem?

A. The schedules for the TaskBot and MetaBot do not intersect

B. Incorrect credentials for the third-party website are being used

C. The input PDF files are corrupt

D. The parameters for the MetaBots variables need to be defined as input variables

ANSWER: D

Explanation:

The parameters for the MetaBots variables need to be defined as input variables. A TaskBot can extract values from the PDF form fields, but those values are not automatically available to a MetaBot when its logic is invoked. To pass the extracted data into the reusable MetaBot component, the relevant MetaBot variables must be configured as input parameters. The calling TaskBot then maps its extracted values to those inputs when it runs the MetaBot. Within the MetaBot, the data-entry logic can use the received input-variable values to populate the appropriate web-application controls. This input/output parameter design is fundamental to how Automation Anywhere v11 supports communication between a TaskBot and MetaBot: the TaskBot supplies runtime business data, while the MetaBot performs the application-specific automation. Correctly defining the variables as inputs ensures that each PDF's extracted field values are passed into the MetaBot for the corresponding third-party-site entry. See Automation Anywhere's MetaBot documentation and the guidance for using MetaBots in TaskBots.

QUESTION NO: 21

How can an administrator pull in some bots into the Control Room which were received via email attachment?

A. Drag-and-drop the Bots into the Control Room > Bots Section

B. Pull in Bots into the Control Room via the Bots > Import Bots command

C. Use Migration to pull in the Bots

D. The administrator would need to copy the Bots to a client and upload the Bots from the client

ANSWER: B

Explanation:

"Pull in Bots into the Control Room via the Bots > Import Bots command" is correct because Automation Anywhere Enterprise v11 Control Room provides an import function specifically for bringing exported bot files into its repository. After saving the bot attachment received by email to an accessible local location, an administrator can open the Control Room's Bots area and use the Import Bots action to select the exported bot package/file. The Control Room then imports the bot into the selected repository location, where it can be managed, assigned appropriate permissions, and made available for execution or further development.

This method is important because bots should be introduced into the Control Room through its managed repository workflow. That workflow enables the Control Room to maintain bot metadata, versioning and access governance consistently. The administrator should also ensure that any required dependent assets, such as associated files or credentials, are handled according to the bot's deployment requirements after import. Automation Anywhere's Enterprise v11 documentation describes Control Room bot-management capabilities and the import process in the Import bots documentation. Additional Control Room bot administration guidance is available in the Bots overview.

QUESTION NO: 22

Which of the following loops are offered by the AAE Client? (Choose three)

A. Database dataset

B. Excel dataset

C. List variables

D. XML dataset

ANSWER: A B D

Explanation:

In Automation Anywhere Enterprise Client, dataset-based looping is used to process structured tabular results one row at a time. **Database dataset** is therefore supported when a bot uses a dataset returned from a database operation and iterates through its records. **Excel dataset** is also supported because worksheet data can be made available as a dataset, allowing the bot to repeat actions for each row while accessing column values through the loop's current-record context. This is a common design for data-driven automations such as updating applications from spreadsheets.

XML dataset is supported for structured XML content. The Loop command can traverse XML data by iterating through nodes selected from the document structure, enabling a bot to extract or process repeated elements such as transactions, users, or line items. These loop types let a bot handle records from database results, Excel-based tabular data, and hierarchical XML content without requiring separately hard-coded actions for every item. Automation Anywhere's Enterprise Client Loop command documentation describes looping over datasets and XML structures; see the Enterprise v11 Loop command documentation and the Enterprise v11 Database command documentation.

QUESTION NO: 23

A Control Room administrator needs to investigate a failed TaskBot execution without changing the TaskBot itself.

Which two Control Room areas can provide execution information for this investigation? (Choose two.)

A. Activity

B. Audit Logs

C. Credential Vault

D. Variable Manager

ANSWER: A B

Explanation:

Activity provides execution monitoring and history information for bot runs. The administrator can review the relevant execution status and identify the failed activity from the historical records.

Audit Logs provide a record of Control Room events and actions. Reviewing audit entries can help an administrator investigate operational activity related to the failed execution, including relevant user and system events.

Activity is primarily used for bot-execution visibility, while Audit Logs provide governance and event-tracking information. Together, they provide useful operational context without requiring changes to the deployed TaskBot. Automation Anywhere documents these capabilities in the Control Room Activity documentation and the Audit Logs documentation.

QUESTION NO: 24

Which of the following WebCR options allows to view the list of all client machines along with their status in read-only form?

A. Clients

B. Security

C. User Management

D. Repository Manager

ANSWER: A

Explanation:

Clients is the Web Control Room area used to view the client machines registered with Automation Anywhere Enterprise and to monitor their current status. In Automation Anywhere v11, a client machine is the device on which Automation Anywhere Client software runs and from which bots can be executed. The Clients view provides a centralized inventory of these machines and displays operational information, including whether a client is available or disconnected. This visibility is important for administrators and bot operators when confirming that an unattended or attended execution machine is reachable before running or scheduling automation.

The question specifically describes a read-only view of all client machines and their statuses, which corresponds to the monitoring/listing purpose of the Clients area in Web Control Room. Automation Anywhere's Control Room documentation identifies client management as the place for viewing and managing registered client devices. See the Automation Anywhere documentation for managing clients and the Enterprise v11 Control Room overview.