UiPath-TCAPv1 Dumps

Shared by Bush on 11-11-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

What is the function of the Rules property in UiPath's Verify Documents Equivalence activity?

Options:

A- It is used to define the formatting rules for the comparison report output
B- It is used to select the pages of the documents to include in the comparison process
C- It is used to exclude specific sections from the document comparison using regex or wildcard-based rules
D- It is used to determine whether images are included in the document comparison

Answer:

C

Explanation:

UiPath Testing Activities support document-equivalence validation by comparing the content of a baseline document with a target document.

Option C is correct because the Rules property allows developers to exclude selected content from comparison through regular-expression or wildcard matching.

This functionality is particularly valuable when testing documents containing dynamic information that changes between executions.

Examples include timestamps, transaction identifiers, document generation dates, and other fields whose values may differ without indicating an application defect.

Developers can define comparison rules through the Create Comparison Rule activity. Each rule specifies a name, a rule type, and a matching pattern.

The supported rule types include RegexRule and WildcardRule.

The resulting comparison-rule objects are supplied to the document-comparison activity through its Rules collection. Matching content is excluded from the comparison, allowing the validation to focus on meaningful differences.

Option A incorrectly associates Rules with report formatting. Option B incorrectly treats it as a page-selection mechanism. Option D describes functionality associated with image or widget comparison settings rather than exclusion rules.

The Rules property retains its fundamental purpose of excluding matching content from comparison.

Official UiPath Reference(s): Testing Activities --- Compare PDF Documents; Create Comparison Rule; Release notes --- Version 25.4.0.

Question 2

Question Type: MultipleChoice

What is the main purpose of a NonProduction runtime in UiPath Orchestrator?

Options:

A- To run unattended processes and verify, they work as expected before deploying them to production.
B- To run test cases created in Test Manager and generate test execution reports.
C- To allow developers to publish packages to Orchestrator feeds without consuming a production license.
D- To execute attended automations in a development environment using Studio's built-in debugger.

Answer:

A

Explanation:

UiPath Orchestrator provides NonProduction runtimes for executing automation processes in unattended mode within development and other non-production environments.

Option A correctly identifies the primary operational purpose of this runtime category.

NonProduction runtimes allow organizations to execute automation processes outside production, supporting validation of process logic, application interactions, configuration settings, and environmental dependencies before deployment.

Unlike attended execution, unattended execution does not require a user to manually trigger the automation through UiPath Assistant.

NonProduction runtime capacity is allocated through Orchestrator and consumed when eligible jobs execute.

Option B describes managed test execution and reporting rather than the primary purpose of NonProduction runtimes. Modern application testing uses dedicated Testing or App Testing runtime capabilities for supported testing workloads.

Option C is incorrect because publishing automation packages is not the execution function of a NonProduction runtime. Option D incorrectly associates NonProduction runtime execution with attended debugging in Studio.

For the Test Cloud Architect examination, the operational distinction remains that NonProduction is intended for unattended non-production process validation, while dedicated testing runtimes provide application-test execution capabilities.

Official UiPath Reference(s): Orchestrator User Guide --- Managing Jobs; Robot Admin Guide --- According to License; UiPath Overview --- Licensing transition.

Question 3

Question Type: MultipleChoice

A tester wants to understand the main function of the Test Explorer panel in Studio.

Which statement correctly describes its purpose?

Options:

A- Displays all reusable templates and allows creation, customization, and scaffolding of test cases
B- Displays all robot and test executions and allows monitoring, logging, and reviewing performance metrics
C- Displays all project dependencies and allows management, configuration, and version control settings
D- Displays all test cases and workflows, and allows grouping, execution, and debugging of tests

Answer:

D

Explanation:

The Test Explorer panel in UiPath Studio provides an integrated interface for managing test execution activities within automation development projects.

Option D correctly describes its primary purpose because Test Explorer enables developers to organize test cases, execute selected tests, debug failures, and inspect testing information.

The panel presents test automation assets according to project type. Within a Process project, both test cases and workflows are displayed by default. Within a Test Automation project, the default view displays test cases.

Test Explorer supports grouping and filtering, enabling developers to focus on relevant portions of the testing portfolio.

Its execution controls allow users to run all visible tests, execute selected test cases, debug individual tests, or rerun previously failed cases.

The panel also supports filtering by execution result, including passed, failed, and not-executed test cases.

For data-driven tests, Test Explorer displays the number of associated data variations and allows developers to manage test data through the context menu.

Option A describes template management rather than the primary Test Explorer functionality. Option B describes execution monitoring and reporting functions more closely associated with Orchestrator and Test Manager. Option C concerns project dependency and source-control management.

Test Explorer therefore serves as Studio's primary testing navigation and execution interface.

Official UiPath Reference(s): Studio User Guide --- Test Explorer; Overview; Working With Test Explorer; Test Explorer actions.

Question 4

Question Type: MultipleChoice

A test architect wants Workflow Analyzer to prevent the commitment of application test workflows that do not comply with required naming and structure conventions.

How should the Application Testing rules be configured to support this objective?

Options:

A- Enable the Application Testing ruleset, adjust rule severities to Error for critical violations, and configure Workflow Analyzer to run automatically before Publish for the test project.
B- Enable only the default Studio ruleset without the Application Testing category, keep all severities as Info, and run Workflow Analyzer manually only when developers remember.

C- Configure only UI Automation selector rules, leave the Application Testing ruleset off, and assume selector validation alone will enforce the required test naming and structure conventions.
D- Disable all built-in Application Testing rules, create a few custom rules in the SDK, and rely on developers to interpret any warnings without enforcing them at publish time.

Answer:

A

Explanation:

UiPath Workflow Analyzer provides static analysis of automation projects and supports enforcement of development conventions before execution, publication, or source-control operations.

Option A correctly identifies the need to enable the Application Testing rules and configure critical violations with Error severity.

Application Testing rules use identifiers beginning with TA. These rules validate characteristics such as test case naming, uniqueness, verification requirements, workflow complexity, and other test automation design conditions.

For example, TA-NMG-002 checks test case naming conventions, while TA-DBP-006 evaluates excessive test case or workflow complexity.

Setting relevant violations to Error allows Workflow Analyzer to treat noncompliance as a blocking condition.

However, Enforce Analyzer before Publish does not prevent Git commits or pushes. It prevents publishing a project containing Error-level violations.

To prevent noncompliant workflows from being committed and pushed to a remote repository, the developer must enable Enforce Analyzer before Push/Check-in under Studio Settings > Design.

Option B lacks mandatory enforcement. Option C ignores the relevant Application Testing rules. Option D replaces built-in validation with an insufficient manual approach.

Official UiPath Reference(s): Testing Activities --- Application Testing Workflow Analyzer Rules; Studio User Guide --- Configuring Studio Settings; About Workflow Analyzer.

Question 5

Question Type: MultipleChoice

Several data-driven test cases share the same Excel worksheet for static customer dat

a. A new column has been added to the worksheet, and its value must be available as an argument in every test case using the shared data set.

Which maintenance approach best aligns with the documented behavior?

Options:

A- Right-click a test case with data variations, choose Update test data, select Existing data, choose the workbook and worksheet, and enable Update all test cases using the same test data before importing.
B- Right-click a test case with data variations, choose Remove test data, select File, choose the workbook and worksheet, and enable Update all test cases using the same test data before importing.
C- Right-click a test case with data variations, open the JSON file directly, add the new column manually, and enable Update all test cases using the same test data before importing.
D- Right-click a test case with data variations, choose Update test data, select File, choose the workbook and worksheet, and enable Update all test cases using the same test data before importing.

Answer:

D

Explanation:

UiPath Studio supports data-driven testing using external datasets, including Excel workbooks. When multiple test cases reference the same dataset, maintaining synchronized data definitions is essential for consistency.

Option D correctly describes the documented procedure for propagating changes made to a shared Excel data source.

The developer begins by opening the Studio project and locating a test case that uses data variations.

From the Project panel, the developer right-clicks the test case and selects Update test data. The appropriate file-based data source is selected, followed by the updated Excel workbook and worksheet.

Studio identifies the available data variations and allows the developer to import the revised data.

Selecting Update all test cases using the same test data applies the updated dataset to other test

cases referencing that source.

UiPath documentation explicitly states that newly added Excel columns are added as arguments to the test case.

Option A incorrectly selects Existing data, which refers to test data already stored within the project. Option B removes the data association rather than updating it. Option C introduces unnecessary manual JSON editing and does not represent the documented shared-update workflow.

After importing the changes, developers should verify the generated arguments and ensure the test logic uses the new data fields appropriately.

Official UiPath Reference(s): Studio User Guide --- Data-Driven Testing; Manage test data; Update test data.

P2P exams

Question 6

Question Type: MultipleChoice

A Test Manager user opens a generated report and wants to investigate systemic issues instead of individual messages.

Which section should the user review first to identify broader recurring themes behind failed test cases?

Options:

A- Common Errors, because it groups similar error messages semantically and is intended to reveal broader systemic categories.
B- Recommendations, because it guides next steps and is intended to categorize failed test cases into broader issue groups.
C- Error Patterns, because it categorizes failed test cases into broader categories that reveal recurring themes and systemic problems.
D- Top Failing Tests, because it lists frequently failed test cases and is intended to classify them into recurring patterns.

Answer:

C

Explanation:

UiPath Autopilot supports test execution analysis by generating reports that identify recurring failures, categorize execution problems, and recommend improvements to test stability.

Option C is correct because the Error Patterns section specifically identifies broader categories of failures that may indicate systemic issues.

Autopilot-generated test reports can contain five sections: Overview, Top Failing Tests, Common Errors, Error Patterns, and Recommendations.

Each section provides a different analytical perspective. Common Errors groups similar error messages to identify frequently occurring exceptions or messages. Error Patterns, however, categorizes failures into broader groups that reveal recurring themes and underlying problems.

For example, several failed test cases may produce different exception messages while sharing a common synchronization or application-availability problem. Error Patterns helps testers recognize this relationship rather than investigating each message independently.

Top Failing Tests identifies test cases with frequent execution failures, while Recommendations provides actions intended to improve testing stability.

Option A incorrectly assigns broader systemic categorization to Common Errors. Option B confuses recommendations with classification. Option D incorrectly describes Top Failing Tests as a failure-pattern analysis mechanism.

The appropriate analytical sequence is to examine Error Patterns, investigate the associated test case logs, and then review Recommendations for suitable corrective actions.

Official UiPath Reference(s): Autopilot User Guide --- Generate test reports; Example prompts for test result analysis.

Question 7

Question Type: MultipleChoice

Which types of technology does Test Cloud support through its broad test automation capabilities?

Options:

A- Web, Desktop, and Mobile with integrated CI/CD and machine learning models testing
B- Only Web, Mainframe and Desktop applications with REST API support
C- Citrix, Web, Desktop, SAP, Mainframe, API, and Mobile
D- Web, Native Desktop, Citrix, SAP, and API limited to CI/CD

Answer:

C

Explanation:

UiPath Test Cloud provides an enterprise testing platform designed to support automated validation across multiple application technologies rather than restricting testing to a particular interface or application category.

Option C correctly lists the broad technology coverage documented by UiPath.

Web application testing enables interaction with browser-based systems. Desktop testing supports Windows applications and other supported desktop interfaces.

Citrix and other virtualized application technologies can be automated using UiPath's UI targeting capabilities, including Computer Vision where conventional selectors are unavailable or unreliable.

SAP testing supports enterprise application workflows, including SAP transaction-based testing and integrations with SAP testing capabilities.

Mainframe automation enables interaction with legacy enterprise applications. API testing supports service-level validation independently of the application's graphical interface.

Mobile testing extends coverage to Android and iOS applications through UiPath Mobile Automation and Appium-based infrastructure.

These technologies can participate in end-to-end test scenarios involving multiple connected systems.

Option A omits several supported technologies and introduces machine-learning model testing as a defining category. Option B incorrectly restricts the supported environments. Option D omits mobile and mainframe testing while incorrectly limiting API capabilities to CI/CD.

This broad coverage allows Test Cloud to support modernization programs involving both legacy and contemporary enterprise applications.

Official UiPath Reference(s): Automation Suite Release Notes --- Introducing Test Cloud; Comprehensive automation coverage; UiPath --- Test Cloud.

Question 8

Question Type: MultipleChoice

What type of information is typically displayed in the Test Explorer panel?

Options:

A- The Test Explorer panel shows all test cases within the test automation project
B- The Test Explorer panel shows only test cases and workflows that have been executed
C- The Test Explorer panel shows only test cases and workflows that have failed
D- The Test Explorer panel shows all the information imported from a data variation source

Answer:

A

Explanation:

UiPath Studio provides the Test Explorer panel as a centralized interface for organizing, executing, debugging, and reviewing automated test cases.

Option A correctly identifies the information displayed by default in a Test Automation project.

According to the official Studio documentation, Test Explorer displays test cases by default when the project type is Test Automation. In Process projects, the default view includes both test cases and workflows.

Test Explorer enables developers to inspect the available test cases, group related tests, and perform operations such as running or debugging individual tests.

The panel also supports filtering based on execution status. Developers can display passed tests, failed tests, and tests that have not yet been executed.

For data-driven test cases, Test Explorer displays the number of associated data variations. These test cases can be expanded to inspect the corresponding variations.

Option B is incorrect because the panel includes tests that have not been executed. Option C incorrectly restricts the display to failed executions. Option D incorrectly characterizes Test Explorer as a complete data-source inspection tool.

A useful distinction is that Test Explorer provides visibility into the project's testing structure, whereas Test Results provides execution outcomes and associated diagnostic details.

Official UiPath Reference(s): Studio User Guide --- Test Explorer; Overview; Working With Test Explorer; Test Explorer actions.

To Get Premium Files for UiPath-TCAPv1 Visit
https://www.p2pexams.com/products/uipath-tcapv1

For More Free Questions Visit
https://www.p2pexams.com/uipath/pdf/uipath-tcapv1

