

advanced qtp interview questions and answers

Advanced QTP Interview Questions and Answers are essential for anyone looking to excel in the field of software testing. Quick Test Professional (QTP), now known as Unified Functional Testing (UFT), is a powerful automation tool used for functional and regression testing of applications. As the demand for skilled QTP professionals grows, so does the need for thorough interview preparation. This article will explore advanced QTP interview questions and provide detailed answers to help candidates demonstrate their expertise and secure their desired positions.

Understanding QTP/UFT Basics

Before diving into advanced interview questions, it's crucial to have a solid grasp of QTP basics. Understanding the fundamentals lays the groundwork for more complex topics.

Key Concepts in QTP

1. Object Repository: A storage place for objects used in your tests.
2. Test Flow: The sequence in which the actions are performed during the test execution.
3. Actions: Modular pieces of code that represent a step in a test.
4. Descriptive Programming: A method to interact with application objects without using the Object Repository.

Advanced QTP Interview Questions

1. What is Descriptive Programming in QTP, and when would you use it?

Descriptive Programming is a technique that allows you to create tests without relying on the Object Repository. Instead of using the predefined properties of objects stored in the repository, you can specify the properties of the objects at runtime.

When to use Descriptive Programming:

- When the Object Repository becomes unwieldy due to a large number of objects.
- When dealing with dynamic objects whose properties change frequently.
- For quick test scripting without the overhead of managing an Object Repository.

2. How do you handle dynamic objects in QTP?

Dynamic objects are those whose properties change during execution. To handle them effectively in QTP, you can use:

- Regular Expressions: Define patterns that match various property values.
- Descriptive Programming: Access objects directly using their properties without relying on the repository.
- Smart Identification: QTP uses a set of properties to identify the object if the primary property fails.

3. Explain the concept of "Synchronization" in QTP. Why is it important?

Synchronization in QTP refers to the process of ensuring that the script waits for the application to reach a specific state before proceeding with the next action. This is crucial because applications may perform operations at different speeds, leading to potential failures in the test scripts if the application is not ready.

Methods of Synchronization:

- WaitProperty: Waits for a specific property of an object to achieve a certain value.
- Sync: Introduces a pause in the test execution for a specified duration.
- Exist: Checks for the existence of an object within a specified timeframe.

4. What is the difference between "Action" and "Function" in QTP?

Actions:

- Designed to encapsulate a sequence of operations.
- Can have parameters and return values, making them modular and reusable.
- Actions can be reusable or non-reusable.

Functions:

- Written using VBScript and can be called in any part of the script.
- They are generally used for performing calculations or processing data.
- Functions cannot have steps that directly interact with the UI elements.

5. How do you implement error handling in QTP?

Error handling is crucial for creating robust test scripts. In QTP, you can implement error handling using:

- On Error Resume Next: Allows the script to continue executing even if an error occurs.
- Error handling functions: Use `Err` object to capture error details and take corrective actions.
- Recovery Scenarios: Define a set of steps to recover from common errors.

Best Practices for Advanced QTP Automation

To ensure effective automation testing with QTP, consider the following best practices:

- **Modularize your tests:** Break down tests into smaller, manageable actions.
- **Use descriptive programming wisely:** Ensure clarity in code by using meaningful variable names.
- **Implement version control:** Track changes to your test scripts for better collaboration.
- **Maintain Object Repositories:** Regularly update and clean your Object Repositories to avoid redundancy.
- **Document your tests:** Provide adequate comments and documentation for easy understanding and maintenance.

Preparing for Advanced QTP Interviews

To excel in advanced QTP interviews, follow these preparation strategies:

1. Practice Coding

Regularly practice writing QTP scripts, focusing on complex scenarios involving dynamic objects, error handling, and descriptive programming.

2. Review Real-World Scenarios

Study real-world testing challenges and how QTP/UFT can resolve them. Understanding practical applications will help you demonstrate your problem-solving skills.

3. Mock Interviews

Participate in mock interviews with peers to refine your responses and gain confidence in discussing advanced concepts.

4. Stay Updated

Keep abreast of the latest updates and features in QTP/UFT, as new functionalities may emerge that could be relevant in interviews.

Conclusion

Advanced QTP Interview Questions and Answers can significantly impact your chances of landing a job in the competitive field of software testing. By mastering the advanced concepts and best practices outlined in this article, candidates can demonstrate their expertise and readiness to tackle the challenges that come with automation testing. Thorough preparation, practice, and a solid understanding of QTP functionalities will ensure that you stand out in interviews and secure a position that aligns with your career goals.

Frequently Asked Questions

What is the difference between QTP and UFT?

QTP (QuickTest Professional) was the original name of the software, while UFT (Unified Functional Testing) is the current name that encompasses both functional and API testing. UFT includes all the features of QTP and adds support for a wider range of technologies and environments.

How can you handle dynamic objects in QTP?

Dynamic objects in QTP can be handled using descriptive programming, where you define the properties of the object at runtime instead of relying on the object repository. You can also use regular expressions to match dynamic property values.

What are the different types of checkpoints in QTP?

QTP supports several types of checkpoints, including Standard Checkpoints, Image Checkpoints, Text Checkpoints, Database Checkpoints, and XML Checkpoints, which allow you to verify the properties of various objects during test execution.

Explain the concept of shared object repository in QTP.

A shared object repository is a centralized storage location for object properties that can be accessed by multiple test scripts. This promotes reusability and consistency across tests, allowing teams to manage object properties more efficiently.

What is the purpose of the recovery scenario manager in QTP?

The recovery scenario manager in QTP is used to handle unexpected events during test execution, such as application crashes or pop-up messages. It allows testers to define recovery scenarios that

can automatically take corrective actions to resume the test.

How do you implement parameterization in QTP?

Parameterization in QTP can be implemented by using data tables to input different sets of data into the test script. You can also use input parameters in function calls or external data sources like Excel files to make the tests more flexible and robust.

What is the significance of the 'sync' method in QTP?

The 'sync' method in QTP is used to pause the execution of the test script until a specified condition is met or until the application is in the desired state. This helps to ensure that the script does not proceed until the application is ready for the next action, improving reliability.

[Advanced Qtp Interview Questions And Answers](#)

Advanced Qtp Interview Questions And Answers

Related Articles

- [algebra 1 review worksheet](#)
- [afls assessment free](#)
- [algebra 2 simplifying expressions](#)

[Back to Home](#)