JSystem

Course 11032 – 32 Hours

Overview

JSystem is an open source test automation framework that is widely used in Israel and around the globe. The framework allows creation of well-designed test automation projects based on layered methodology. The framework is written in Java and allows creation of infrastructure and building blocks for all types of users.

On Completion, Delegates will be able to

- Understand the structure of test automation project
- Understand the differences between test automation framework and tools
- Have the ability to design a test automation software solution
- Create infrastructure that will allow communication with the devices under test
- Crate building blocks that will allow creation of test automation test cases

Who Should Attend

- Test automation engineers that want to create a test automation project or to join test automation team.

Prerequisites

- The course requires basic programming skills in object oriented language like Java, C# or C++
- Knowledge and background in test automation.

Course Contents

Introduction to Test Automation

- The benefits of open source software
- Selecting the right tool for the job
- Test automation frameworks overview
- Peripheral systems overview

Introduction to JSystem

- The layered architecture
- Project example

Java and Eclipse Basics

- Introduction to Eclipse
- Java – Write once run everywhere
- Variables and operators
- Flow control
- Methods, objects and classes
- Exceptions
- Java collections

Introduction to Maven

- Project life cycle
- GAV
- Dependency management
- Plugins
- Archetypes

Introduction to JUnit

- A little bit history - Kent Beck
- TDD methodology
- Create a simple project using maven and practice the use of JUnit as unit-tests framework and the TDD methodology.

Creating JSystem projects using Maven

- Create a SO and tests project and link them together
- What happens in the project build process?

JSystem Tests Layer

- Building block creation and parameterization
- Add information to building blocks
- Advanced parameters
- References
- Dynamic parameters
- Scenarios

Reporters

- Understanding JSystem report mechanism
- The JSystem default reporters
- Implementing reporters

Remote debugging building blocks using Eclipse

- Understanding the JSystem architecture
- Configuring the JSystem properties
- Configuring Eclipse
- Using the debug check box

System Object Life Cycle

- The benefits of using system objects
- System objects hierarchy
- System object life cycle. Init and close
- JSystem class loader
- The object manager

Analyzers - Analyze the results using JSystem

- The motivation behind the analyzers design pattern
- Overview of all the existing JSystem analyzers
- Implementing new analyzers

Cli Connection

- Using the CliConnectionImpl, CliCommand and Prompt classes
- Connecting the SSH, Telnet and RS232 devices
- Configuring the system object using the SUT configuration file

Connectivity Packages

- Core drivers (SNMP, TCL, Vb, Python, Shell, etc...)
- Network drivers (IXIA, TestCenter, iPerf, etc…)
- UI drivers (Selenium, Jemmy, etc..)

Fixtures

- The motivation behind test fixtures
- Linking fixtures to test classes and scenarios
- Implementing fixtures

Monitors

- The motivation behind monitors
- The monitors manager
- How to use monitors
- Implementing monitors

Jenkins

- Introduction to Jenkins
- Integration of JSystem with Jenkins.