Chapter 1

Introduction

➢ **In this chapter…**

1.1 The Purpose of this Document

This Developer and User Guide describes the JSystem Automation Framework, the complete testing solution, enabling the user to write drivers, tests and build test scenarios. These scenarios can then be executed and viewed and the test data can be published using the Reports application or by publishing in-depth reports by using JReporter.

The following chapters describe the JSystem software specifications, operating parameters and application configuration. The advanced developer coding requirements and test integration as well as scenario structuring and a detailed trouble shooting guide.

1.1.1 Introduction to JSystem Automation Framework

JSystem is a framework for writing and running automated tests; the **JSystem Automation Framework** is written in Java and based on open source java projects and custom code.

The main goal of these projects is to support automation of functional and system testing.

JSystem Automation Framework supports the full testing lifecycle, providing a solution for various types of users. JSystem enables the user to start with a small set up performing simple tests, and then enlarge the solution, providing a flexible solution for growing client needs. The JSystem Automation Framework is based on Java using Eclipse as the development environment.

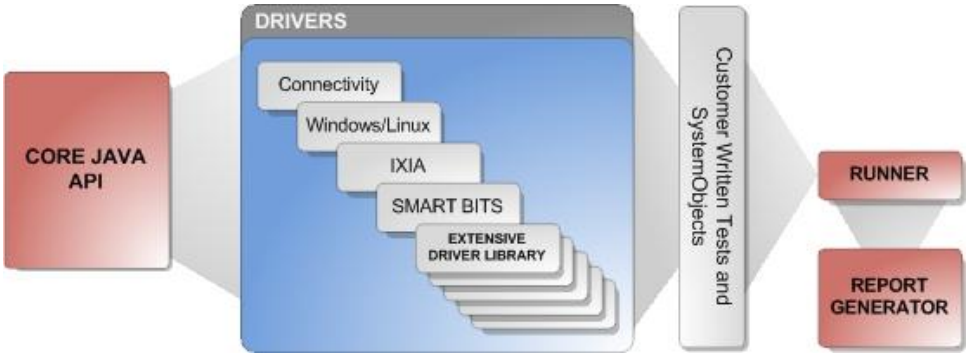

Figure 1: JSystem Automation Framework Components

The flow diagram above illustrates the relationship between the various components within the JSystem Automation Framework.

1.1.2 JSystem Automation Framework

JSystem delivers a Framework for system testing automation that provides a model and architecture structure that supports complex testing environments. This enables the user to build tests with full setup configuration independence, define test parameterization, build complex scenarios and utilize enhanced reporting functionality.

1.2 JSystem Testing Approach

The JSystem approach to today's testing requirements makes a clear distinction between the graphic user interface (GUI) and the business logic, testing the business logic as a separate entity from the user interface.

The JSystem solution is built on a committed API. By focusing on the business logic JSystem enables the user to take full advantage of the automated testing environment, providing increased testing stability and simplifying the testing project for the end user.

When dealing with an automation project JSystem assumes that the automation is a software project. The JSystem testing framework is augmented with a dynamic methodology and a robust architecture solution. The JSystem drivers enable the user to connect to multiple devices commonly found in a typical lab environment.

1.2.1 Current Method of Testing

The drawing illustrates how independently written custom drivers are required for each peripheral object and or item, this is both labor intensive and repetitive increasing the overall work load and complexity of the testing environment.

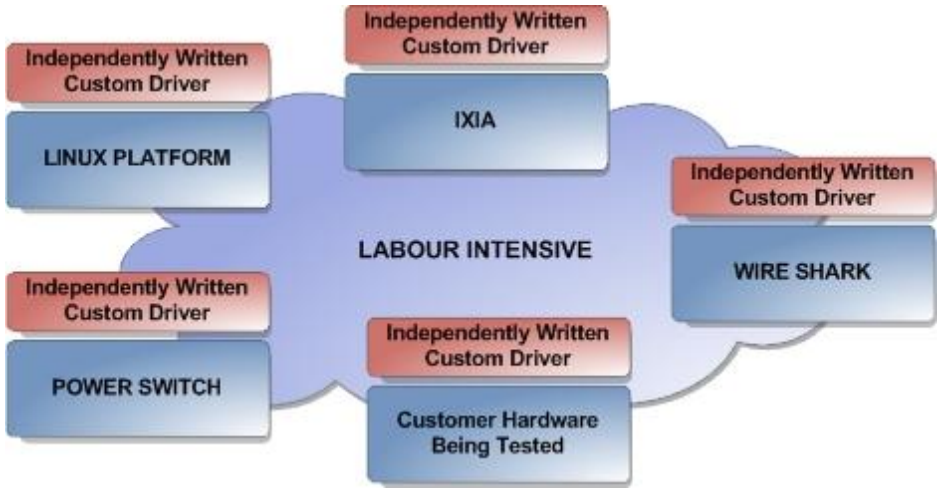

Figure 2: Current Testing Complexity Problem

1.3 JSystem Automation Framework Solution

The JSystem Automation Framework offers a total out of the box solution enabling complete system integration with all hardware and software peripherals. As the drawing clearly illustrates the JSystem Automation Framework centralizes the functionality and provides the user with a single location where all maintenance is performed.

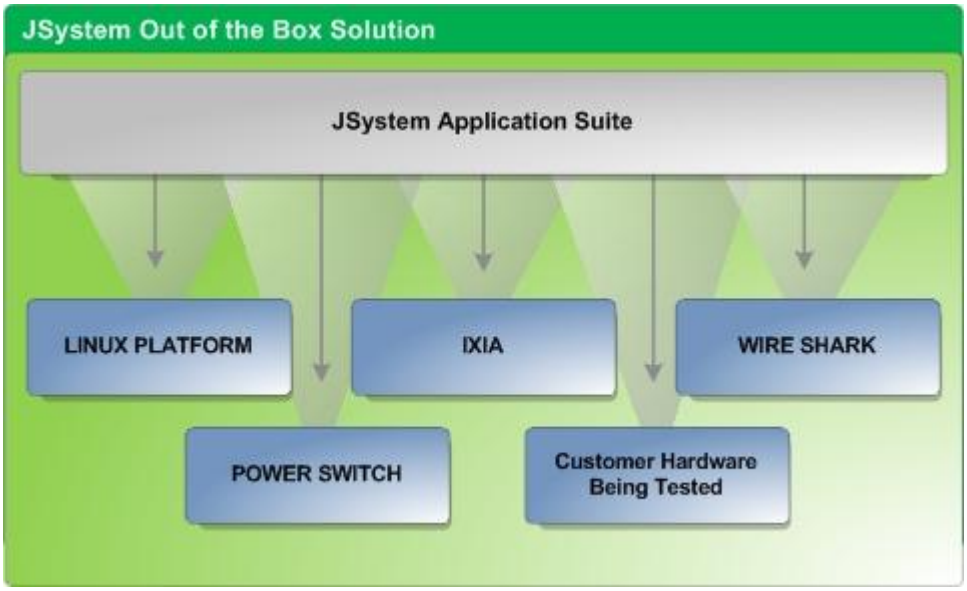

Figure 3: JSystem Automation Framework Solution

1.3.1 JSystem Key Focus

The central issue the JSystem Automation Framework solves is the maintenance aspect of the automation project providing the user tools to keep project maintenance to a minimum.

JSystem does this by focusing on four key aspects:

- **Maintainability** – JSystem enables the user to adjust the automation changes via a modulated system referred to as system objects (SystemObjects), These **SystemObjects** communicate directly to the business logic of the product. JSystem has the ability to connect directly to the application API enabling low maintenance migration.
- **Visibility** – JSystem provides a tool set that enables all user profiles the ability to easily interact with each other by clearly aligning the level of information they require. This stream lines the testing process by displaying the relevant information to each user profile.
- **Scalability** – The ability for a test project to grow in scale from ten's of tests, to hundreds of tests to, thousands of tests. The JSystem application suite is built on an expandable code foundation that envisions project scaling and growth from the outset of the testing project.
- **Simplicity** – By clearly defining the user profiles level of understanding the system divides the test project into layers. These layers offer simplified environments for each user profile.