Open2Test Test Automation Framework for SilkTest – FAQ

Version 1.0

January 2010

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to answer frequently asked Questions about the Open2Test Test Automation Framework for SilkTest.

2 Frequently Asked Questions

2.1 Open Source Test Automation

The table below lists FAQs, along with their respective answers, related to the Open2Test Test Automation Framework for SilkTest.

1.	Is the Open Source Test Automation Framework application-dependent? **No, it is not application-dependent. The Open Source Test Automation Framework is designed to perform operations for all standard object types (JavaJFCPushbutton, JavaJFCCombobox, JavaJFCCheckbox, etc.) of technology commonly used in Java application.**
2.	Is the Open Source Test Automation Framework technology-dependent? **Yes, it is technology-dependent.**
3.	What are benefits of this framework? **The Open Source Test Automation Framework provides the following benefits:** **1. The testers can automate test cases without the help of programmers or programming background.** **2. They can run automated test cases more reliably.** **3. The Open Source Test Automation Framework reduces maintenance and increases productivity.** **4. Keywords are application-independent.**
4.	Does the Open Source Test Automation Framework support a data-driven framework? **Yes.**
5.	Does it support a function modular framework? **Yes.**
6.	What are the prerequisites for using the Open Source Test Automation Framework? **The prerequisites for using the Open Source Test Automation Framework are:** **1. Knowledge of the keywords present in the Keyword reference document.** **2. Basic knowledge of the SilkTest tool.** **3. All the necessary files should be associated.**
7.	Does the Open Source Test Automation Framework support all versions of SilkTest? **Almost. For more information, please refer to the support matrix document.**
8.	What are the various components of the Open Source Test Automation Framework? **The components include: Driver script, function library, common functions, object repository, keywords, external test data, and global variables.**
9.	What are the different components needed to run a SilkTest script using the Open Source Test Automation Framework? **Include the following list of files inside the data-driven testcase (.g.t file):** **1. Open Source Test Automation Framework.**

	2. 'datadrivetc.inc' file. **3. Object repository file.** **Then call the statement mentioned below in the data-driven test case to call the Open Source Test Automation Framework:** **Keyword_Driver('<Parameter1','<Parameter2>','<Parameter3>', '<Parameter4>','<Parameter5>')**
10.	Does the Open Source Test Automation Framework support external functions? **Yes, it supports external functions.**
11.	How does the Open Source Test Automation Framework work? **Using the Open Source Test Automation Framework, testers can develop test cases using Microsoft Excel and the available list of keywords. When the test is executed, the framework processes the Excel workbook and calls the functions associated with the keywords entered in the Excel spreadsheet. These keyword functions in turn perform specific actions against the Application Under Test (AUT).**
12.	What are the benefits of the Open Source Test Automation Framework? **Reusability, greater test productivity, optimum utilization of the tool through keyword support, and minimum effort needed to build scripts.**
13.	What are the features of the Open Source Test Automation Framework? **The following are the features of the Open Source Test Automation Framework:** • **Performing operations and verifications on the objects** • **Usage of variables** • **Conditional checking** • **Data-driven testing** • **Reports** • **Exception handling** **For more information, please refer to the Support Matrix documents.**
14.	How reliable is this framework as compared to a linear script? **The framework provides standardized, tested code. It is typically much more reliable and more thoroughly tested than recorded scripts. It also provides uniformity across automation scripts and ensures standard procedures are followed for coding.**
15.	What are the advantages of building scripts using the Open Source Test Automation Framework over writing code in SilkTest?

	Because this is a keyword-driven framework, the user does not need to know programming in 4Test Language. Testers can develop scripts without learning the underlying automation tool. The tests are easier to understand and maintain, and they provide maximum code reuse.
16.	What is the Open Source Test Automation Framework? **This framework is built for several of the leading test automation tools. It allows scripting of test cases using a set of keywords provided as part of framework. This is often referred to as keyword-driven testing or action-based testing.**

2.2 Keyword Scripts

The table below lists FAQs, along with their respective answers, related to keyword scripts.

1.	What are the various keywords available in the Open Source Test Automation Framework? **Refer to the Keyword Reference Document for the list of keywords that are available in the Open Source Test Automation Framework. The document covers all the keywords, as well as their syntax and usage.**
2.	Does the Callfunction keyword support variables as arguments? **Yes. Please refer to the Framework Extensibility document for more information.**
3.	Does the Open Source Test Automation Framework support if-else conditions? **Yes. Use the Condition keyword.**
4.	Why doesn't the Window close even after using the 'Close' keyword? **Before closing the Window, set the context first.**
5.	Does the Open Source Test Automation Framework support user-defined functions? **Yes. Please refer to the Keyword Reference document for the keyword to be used.**
6.	I am getting a 'Numeric Field overflow' error when I use a wait statement in the keyword script. **Please use a single quotation mark before the numeric value. For example: '1.5.**