INTRODUCTION

Silk Test is a tool specifically designed for doing REGRESSION AND FUNCTIONALITY testing. It is developed by Segue Software Inc. Silk Test is the industry’s leading functional testing product for e-business applications, whether Window based, Web, Java, or traditional client/server-based. Silk Test also offers test planning, management, direct database access and validation, the flexible and robust 4Test scripting language, a built in recovery system for unattended testing, and the ability to test across multiple platforms, browsers and technologies.

You have two ways to create automated tests using silktest:

1. Use the **Record Testcase command** to record actions and verification steps as you navigate through the application.
2. Write the testcase manually using the **Visual 4Test scripting language**.

1. Record Testcase

The Record / Testcase command is used to record actions and verification steps as you navigate through the application. Tests are recorded in an object-oriented language called **Visual 4Test**. The recorded test reads like a logical trace of all of the steps that were completed by the user. The Silk Test point and click verification system allows you to record the verification step by selecting from a list of properties that are appropriate for the type of object being tested. For example, you can verify the text is stored in a text field.

2. Write the Testcase manually

We can write tests that are capable of accomplishing many variations on a test. The key here is re-use. A test case can be designed to take parameters including input data and expected results. This “data-driven” testcase is really an instance of a class of test cases that performs certain steps to drive and verify the application-under-test. Each instance varies by the data that it carries. Since far fewer tests are written with this approach, changes in the GUI will result in reduced effort in updating tests. A data-driven test design also allows for
the externalization of testcase data and makes it possible to divide the responsibilities for developing testing requirements and for developing test automation. For example, it may be that a group of domain experts create the Testplan Detail while another group of test engineers develop tests to satisfy those requirements.

In a script file, an automated testcase ideally addresses one test requirement. Specifically, a 4Test function that begins with the test case keyword and contains a sequence of 4Test statements. It drives an application to the state to be tested, verifies that the application works as expected, and returns the application to its base state.

A script file is a file that contains one or more related testcases. A script file has a .t extension, such as find .t

**Other Segue products**
The Silk products include

- **Sit Test** for functional and regression testing
- **Silk Performer** for load and performance simulation
- **Silk Pilot** for functional and regression testing CORBA and EJB servers
- **Silk Radar** for automated defect tracking
- **Silk Vision** for enterprise application health monitoring
- **Silk Express** for a scalability and load testing consulting solution.

**Silk Test Features**:
Some of the feature of silk test are given below.

- Easy to use interface
- Built in recovery system
- The object oriented concept
- Record & Play
- Multi-kind application testing
- Automatic generation of results
- Browser & Platform independent
- 24 x 365 unattended testing
- Distributed Access to Test Results
- Cross – Platform Java Testing
- Testing Across Multiple Browsers and Windows Versions
- Support for HTML, XML, JavaScript, Java, Active X, Windows controls, and Visual Basic.
- Single-recording testing for cross-platform Java testing with the Silk Bean.
- Against Over 35 Databases.
- Link Tester
- Validation of Advanced Database Structure and Techniques
- Creation of
  - Test Plan
  - Test Frame
  - Test Suite
- Integration with other silk products.

Silk Test Architecture

Normal use of an application consists of a person manipulating a keyboard and mouse to initiate application operations. The person is said to be interacting with the GUI (Graphical User Interface). During Silk Test testing, Silk Test interacts with the GUI to submit operations to the application automatically.

Thus Silk Test can simulate the actions of a person who is exercising all the capabilities of an application and verifying the results of each operation. The simulated user (Silk test) is said to be driving the application. The application under test reacts to the simulated user exactly as it would react to a human rest. Silk Test consists of two distinct software components that execute in separate processes:

The Silk Test host software
The 4Test Agent software
Silk Test host software

The Silk Test host software is the program you use to develop, edit, compile, run and debug your 4Test scripts and test plans. This manual refers to the system that runs this program as the host machine or the Silk Test machine.

The Agent

The 4Test Agent is the software process that translates the commands in your 4Test scripts into GUI-specific commands. In other words, it is the Agent that actually drives and monitors the application you are testing. One Agent can run locally on the host machine. In a networked environment, any number of Agents can run on remote machines. This manual refers to the systems that run remote Agents as target machines. This manual refers to the systems that run remote Agents as target machines. In a client/server environment, Silk Test drives the client application by means of an Agent process running on each application’s machine. The application then drives the server just as it always does. Silk Test is also capable of driving the GUI belonging to a server or of directly driving a server database by running scripts that submit SQL statements to the database. These methods of directly manipulating the server application are intended to support testing in which the client application drives the server.

Limitations of Silk Test:

Some of the limitations of Silk Test are given below:

- SilkTest may not recognize some objects in a window / page due to some technical reasons.
- Silk Test may not recognize some window frames.
- The ‘tag’ value may get changed frequently.
- Sometimes it will be difficult to activate some window.
- It may be necessary to make some modifications if testing should be shifted to other browser / operating system.
In web based applications, sometimes, silktest will take the links as simple text.

**System Requirements:**

The minimum requirement, a system needs to run the silk test is given below:

- Windows NT, Windows 95, 98 or 2000
- Pentium 466 Mhz or better processor (application dependent)
- 32 MB RAM
- 60MB Hard Disk

**Supported Environments:**

- Netscape Navigator 4.x
- Internet Explorer 4 and 5
- Active X, Visual Basic 5 and 6
- Java JDK 1.3
- Swing 1.1
- Microsoft Web browser control

**THE AUTOMATED TESTING PROCESS**

The testing process has these for steps:

- Creating a testplan (if you are using the testplan editor)
- Recording a test frame.
- Creating testcases
- Running testcases and interpreting their results.

**Creating a testplan**

If the testplan editor is used, the automated testing process is started by creating a testplan. A basic testplan is structured as a hierarchical outline and contains:
• Descriptions of individual tests and groups of tests. As many levels of description can be used.
• Statements that link the test descriptions in the plan to the 4Test routines, called testcases, that accomplish the actual work of testing.

Recording a test frame

Next, record a test frame, which contains descriptions, called window declarations, of each of the GUI objects in your application. A window declaration specifies a logical, cross-platform name for a GUI object, called the identifier, and maps the identifier to the object’s actual name, called the tag. In addition, the declaration indicates the type of the object, called its class.

Creating testcases

The 4Test commands in a testcase collectively perform three distinct actions:
• Drive the application to the state to be tested.
• Verify the state (this is the heart of the testcase).
• Return the application to its original state.

The powerful object-oriented recorder can be used to automatically capture these 4Test commands to interact with the application, or to write the 4Test code manually if one is comfortable with programming languages. For maximum ease and power, these two approaches can be combined, recording the basic testcase and then extending it using 4Test’s flow of control features.

Running testcases and interpreting results

Next, run one or more testcases, either by running a collection of scripts, called a suite, or, if you are using the testplan editor, by running specific portions of the testplan. As each testcase runs, statistics are written to a results file. The results file and its associated comparison tools allow you to quickly pinpoint the problems in your application.
A Test Frame

The test frame is the backbone that supports the testcases and scripts. It is a file that contains all the information about the application’s GUI objects that Silk Test needs when you record testcases. This information minimally consists of a declaration for each GUI object, but can also include any data that you want to associate with each GUI object, as well as any new classes and methods that you want to define.

A window declaration specifies a cross-platform, logical name for a GUI object, called the identifier, and maps the identifier to the object’s actual name, called the tag. Because the testcases use logical names, if the object’s actual name changes on the current GUI, on another GUI, or in a localized version of the application, only the tag in the window declarations need to be changed; don’t need to change any of the scripts. Variables, functions, methods, properties can be added to the basic window declarations recorded by Silk Test.

To record declarations for the main window and menu hierarchy of your application:

1. Start up your application and Silktest.
3. Select the Test Frame radio button and click OK. The New Test Frame dialog is displayed, allowing to create a test.
4. Frame file for an application displayed in the Application list box.
5. Select the application from the Application list box. If a Web application is tested, different fields are seen.
6. Click OK. The new test frame file is created. The file contains the 4Test declarations for the main window and all its menus, as well as a generic declaration that is valid for each of the standard message boxes in the application.
A Test Plan

A Testplan is made up of a large amount of information, a structured, hierarchical outline provides an ideal model for organizing and developing the details of the plan. A testplan consists of two distinct parts

- An outline that describes the test requirements
- Statement that connect the outline to the 4Test scripts and testcases that implement the test requirements.

Using the testplan, we can create and run tests.

To start a new testplan:

- Select File / New
- Select Testplan and click OK.

An empty testplan window opens.

A Test Suite

A Test Suite is a collection of test scripts. Consider a case that we are having a set of script (.t) file. If we want to run these scripts against our application, we have to select the required testcase or we have to run the entire script file. But after the completion of that script file, the user has to manually change that to the next script file to run those testcases available in that script. Instead of that silktest provides a way to continuously select a set of script files and run those script files at-a-stretch. This can be done by creating a new Test Suite file and declare the needed script files in that suite file.

To start a new test suite:

1. Select File / New.
2. Select Test Suite and click OK.
3. In that suite file enter the script file names to run continuously.
4. Save that script file.
5. Compile the script file and run it.
6. Now the process of running the script will not stop after the completion of the first script file, instead of that it will automatically pass to the next script file and run the testcases available there.

Assume a case where there is a folder called silk scripts in c drive with five test script files. Here in the suite file, we are calling all the script files instead of running those script files separately. The suite file will look like as given below:

```plaintext
// Clicknglinks.t
use c:silkscripts.script.t"  
use “c:silkscripts.script2t”
use “c:silkscripts./script3.t”
use”c:silkscripts\script4.t”
use”c:silkscripts\5.”
```

**A Test script**

A testscript file contains various testcases for various test conditions.

**A Testcase**

In a script file, a testcase ideally addresses one test requirement. Specifically, a 4Test function that begins with the testcase keyword and contains a sequence of 4Test statements. It drives an application to the state to be tested, verifies that the application works as expected, and returns the application to its base state.

- In the silktest tool, select the File -> New option the menu bar.
- In the resulting dialog box ‘New’, there will be options for selecting different kind of files.
- Select the file type 4’ test script’ option.
- It will open a new script file.
Before start writing the testcase, declare the necessary file that is to be used in that script file.

Start with the keyword ‘testcase’ followed by the testcase name. The name of the testcase is whatever as selected by the user. But make sure that by looking at the name of the testcase, the objective of the testcase should be understandable.

Start the tests from the scratch so that the silktest will start the application and to the testing from the base state.

Use necessary conditions / loops if necessary.

At the end of each and every script, print a statement to know whether the test case has achieved its objective or not. The user can make sure that the particular part of the application is error free by looking at the message you print.

Try to make the testcase effective and time consuming (say) by keeping the second tests continue from the place where the first test finishes.

A sample testcase for registering into the yahoo mail.

Testcase registration ()

- Browser.LoadPage “mail.yoo.com”)
- Sign In YahooMail.SetActive ()
- SignIn Yahoo.Mail.objSingIn YahooMail.SignUp.Now.Click()
- Sleep (3)
- WelcomeTo Yahoo.Set Active
- Welcome To yahoo.objWelcomeToYahoo.LastName.SetText(“lastname”)
- Welcome To Yahoo.objWelcomeToYahoo.LanguageConten1.Select(5)
- WelcomeTo Yahoo.objWelcome To Yahoo.ContactMeOccassionally About.Click ()
- Welcome To Yahoo.objWelcome To Yahoo.Submit ThisForm.Click()
- If Registration Success.Exists ()
- Print (“Test Pass”)
- else
- logerror (“Test Fail”)

SilkTest Tutorials → Sanjay Kumar
Silk Test installation tips

- Run the Silk Test setup from the CD or through the Network.
- The Silk Test Software is available in the Firesip\Europa\software\siltest5.0.1\siltest’ directory in your Network Neighborhood.
- Get into the above folder and select the setup.exe file to start the installation.
- During installation, it will ask for the licence file. Set the path “Firesip\Europa\software\siltest 5.0.1\licence” for the “licence.dat” file.
- In the installation process, it will ask for the Silk/test/Silk Test Agent only option. Select the Silk Test option if you are installing this for testing applications in the stand-alone machine.
- For the ‘Will you be testing browsers?’ message box, select “Yes” if you are going to test web based applications.

- It will ask for the default browser option. Select the appropriate Browser you want to test the application using Silk Test. Note that you are allowed to select only one Browser option. By default Silk Test goes fine with Netscape browsers.
- After installing, it will open the Silk Test tool with the Quick start wizard open. The quickstart wizard will assist you in creating various silk files. If you are a first time user of silktest, the continue with that.

Getting started with the Quickstart Wizard

when you are using Silk Test with the testplan editor, you can use the QuickStart Wizard, which greatly simplifies the four steps of automated testing.

hen you start Silk Test the first time (or whenever you start and have no open windows), the QuickStart Wizard is displayed automatically. You can also invoke the wizard at any time by selecting File/New and clicking the Quickstart Wizard icon.
You can use the Quickstart wizard to:

1. **Create a testplan**
   
   You simply name the file (giving it the .pln extension) and its directory.

2. **Create a test frame,**
   
   Which contains descriptions of the GUI objects in your application that you want to test. As prompted, you simply open your application and open the various windows and dialogs that you want to test in the application. The wizard automatically records all the declarations in a file called frame.inc. You don’t have to do any coding.

3. **Record testcases**
   
   You name the testcase and provide a description for the testplan, then simply record the testcase. Again, you don’t have to do any coding. The wizard automatically saves the testcase in script (.t) file with the same name as the testplan.

**Run testcases.**

**Procedure To use the wizard:**

1. Involve the wizard by selecting File /New and clicking the QuickStart Wizard icon. Now you will name a new testplan, which will organize and manage your tests.
2. Click Next.
3. Name the file edit.pln and Next. The next step is to record the test frame, which defines all the windows, dialogs, menus, and so on that you want to test.
4. To create a new test frame, leave New Test Frame selected and click Next. At this point, the wizard lists all the open (running and not minimized) applications. If Text Editor is not open, you can open it now (it is the directory where you installed Silk Test). After you open the Text Editor, click on the QuickStart Wizard title bar to see Text Editor added to the list of applications.
5. Select Text Editor and click next.
6. The capture Windows panel displays, describing the procedure.
7. Click Next.
8. Now you simply open a document window and open all the dialogs that you want to test in the Text Editor. When you place the mouse pointer on a window or dialog, the wizard records all the declarations that SilkTest needs in a file called frame.inc in the same directory as your testplan.
9. When you have finished capturing the windows and dialogs in Text Editor, click Return to Wizard in the Capturing New Windows dialog. Now that you have created your test frame, you are ready to create a testcase.
10. Click Next twice.
11. Name the test Find Box and enter the description “Verify controls in Find dialog.” Click Next. Your test is now being recorded, as indicated by the Record Status window on your screen.
12. Now go to Text Editor, select Search / Find to open the Find dialog, place your mouse pointer over the dialog’s title bar, and press Ctrl + Alt to verify its state. The verify window displays. Click OK to verify all properties for the dialog. Close the Find dialog (to return to your base state), then click Done in the Record Status window. You return to the Wizard and are asked to confirm that the test is what you want.
13. Click Next.
14. Run the test by clicking the Run Test Button.
15. The wizard reports the results. You can move the wizard to the side and look at the results file that is created whenever you run a test.
16. In the wizard, click Next to save your testcase. The testcase is saved in a script (.t) file with the same name as testplan (in this case, edit.t)
17. Click Close to close the wizard. You see a window containing the results file from the test you just ran. In another window is the testplan.

Configuring the settings

- In Silk Test, select Option -> Extentions menu from the menu bar.
- It will load the Extentions dialog box.
- In that dialog box, check the kind of application you're testing.
- Say, if you are testing the web-based application in the Netscape browser, Enable the Netscape option by checking against it and un-check all the other options.
• Click on the OK button.
• Now click on the Options - > Runtime option.
• Set the ‘use path’ option to point to the silktest installed folder. Say if the silktest has been installed in you c:\ drive then set the sue files = “C:\Program Files\Segue\SilkTest\ETEND”. This is to use the common include files as per the application.
• Since we have selected the Extensions, the use file will have the declaration ‘extend”\netscape.inc’ of the include file. Now the script is ready to open the Netscape browser by default and run the scripts.
• From the status bar Go to the Start - > Programs - > Silk Test - > Extention Enabler option, to declare the same set of extensions (as in step 4) in the ‘Extention Enabler’ dialog box.

Exposure to Silk Test IDE

To Start the Silk Test tool
• Select the ‘Start’ button in Windows status bar.
• Go to Program - > Silk Test - > Silk Test’ option (with green color icon)
• It will open the silktest tool.
• If you get the Quickstart wizard, close the wizard if you don’t need it or else, refer to the ‘Installation tips’ to go with the wizard.

To compile & run scripts
• Open the testsscript file (with the extension .t).
• Select the Run - > Compile option from the menu bar. Also we can select the icon with the ‘tick’ mark to compile the scripts.
• The files that are associated with the script files will also get compiled. Hence if there is any problem with the include file, you will get error by compiling the testscript file itself.
To compile a selected file, keep the mouse cursor on that particular file and give the Run-Compile option.

The ‘Run option will get enabled only for the testscript files. We cannot run a include file or a test frame.

We can run a testcase selectively, compile the testscript file and click the “=>t” icon in the tool bar.

To run the entire set of testcases in a testscript file, click the “==>=” icon or select the Run - > Run option.

If there is any syntax error in the script file, it will show that if you move to the next line of the code. But the other errors are known only at the time of compilation or during the run.

To open a new or existing files

In Silktest,

- Select the File - > New menu from the menu bar. (or) select from the ‘white icon’ in the left top corner of the window.
- It will ask for various kind of files to open. Select the ‘4Test include file’ to declare the window objects from the application.
- The ‘4Test Script file’ option is to open the script file where we will be writing testscript.
- The above two files are more important for building the scripts.
- Open the application you want to test. If you are going to test the Yahoo site, then open the browser load the page you want to start testing.
- The page that you start testing the application will be assumed as the ‘BaseState’. Even we can explicitly declare the window base state.
- The Test Script file will be used only after creating the include file. We will be using the include file to write the script file. Hence we have to declare the include file that we are calling in the testscript files.
- To open an existing file, select File - > Open and select the existing file.
To write a new plug & play testcase (using Action)

- This example is to write the script for logging into the yahoo site.
- Start the Silk Test by selecting from the ‘Start’ menu.
- Configure the settings as given in lab I.
- Click ‘File -> New ‘menu, and select the ‘4Test script file’ option.
- Click on the OK button.
- Start with the keyword ‘testcase Action ()’ and press Enter key in your keyboard. The testcase is the default keyword for any testcase and the name Action is the name of the testcase. The testcase name can be anything but it is advisable to name it clearly so that will represent the functionality of the test.
- Now start writing the testcase (Follow the below instructions)
- Open the application in parallel ie., open the browser in which the application has to run. (say Netscape)
- Go to Silk Test
- Click Record -> Actions. Menu from the menu bar.
- It will load the ‘Record Actions’ dialog box.
- Keep the dialog box as it is, go to the application and do the action what ever you want to perform.
- The silktest will record the events you do sequentially and you can vie it in the ‘Record Actions’ dialog.
- After completing your task (till whatever you want to record), click on the ‘paste to editor’ button in the ‘Record Action’ dialog box.
- Then click the close button to close the Record actions dialog box and go to your application.
- Now the recorded code will be readily available in the testscript editor, inside the testcase.
- Now delete the keyword ‘recording’ in your first line of the recorded code.
- Now, select the entire recorded code by keeping the mouse arrow at the leftmost dot (.) in your editor at the first line, and drag it till the end.
- Right click on the selected code and select the ‘Move Left’ option.
- The code is ready now.
- Now, compile the code from the ‘Run -> compile’ option and run the script by selecting the ‘Run -> Run’ menu.
• Now the testcase will automatically start the application and perform the events recorded and throws the results.

• The sample recorded testcase for yahoo login look like this:

• Testcase Action()

  ▪ // [-] recording
  ▪ Browserpage.Set Active()
  ▪ Browser.Location.Set Text (www.yahoo.com”)
  ▪ Browser.Location.Type Keys (“”)
  ▪ Yahoo.HtmlLink (“Mail| #26|http://www.yahoo.com/r?m2”).Click()
  ▪ BrowserPage.HtmlTextField(“Yahoo!ID: | # 1”). Set Position (1, 1)
  ▪ BrowserPage.HtmlTextField(“Yahoo!ID: | # 1”). Set Text (“username”)
  ▪ BrowserPage.HtmlTextField(“Yahoo!ID: | # 1”). TypeKeys (“”)
  ▪ BrowserPage.HtmlTextField(“Password:|#2”).Set Text (“password”)
  ▪ BrowserPage.HtmlPushButton ( “Sign In|#1”). Click()

The 4 Test language

The 4 Test language is an object-oriented fourth-generation language (4GL) designed specifically with the needs of the QA professional in mind. 4 Test’s powerful features are organized into three basic kinds of functionality:

• A robust library of object-oriented classes and methods that specify how a testcase can interact with an application’s GUI objects.
• A set of statements, operators, and data types that you use to add structure and logic to a recorded testcase.
• A library of built-in functions for performing common support tasks. Note This section provides a high-level look at 4 Test.

Running the silkscripts

The basic silk scripts will be in two forms. One as an include file and the other as a script file.
The include file with the extension *.inc can be used for the declaration of window names, window objects, variables, constants, structures and classes. The core objects of the scripts lies here.

The script file will be used in writing scripts. It will be with the extension *.t. The body of the scripts will be defined here. ie. the testcases that meets various test conditions will be written in the script file.

The script file (*.t) can be used for declaring objects and the include file (*.inc) for writing testcases. But to make the code clear we use different files for different purposes. If no testcases is written in a file (include), then the include file can be compiled but cannot be run. It will show error that the file does not contain any testcases. Only the file with the testcase present will be allowed to run.

Before running the scripts, separate declaration file have to be written (for declaring the objects) and the script file (for writing scripts using that declaration file) and compile them. The steps to be followed for running the scripts are as below.

- Open the silk test tool.
- Open the script (*.t) file that has to be run.
- Compile the script by selecting the Run-> Compile menu from the menu bar (or) from the compile icon.
- It will compile that particular script and the other related files, called by that script. The user can confirm that by looking at the progress status (in yellow color) in the bottom-right corner of the silktest too.
- If there is any error, the error details are displayed in the compile time. The user has to make necessary changes.
- Then, select the Run-> Testcase from the menu bar (or) else select the Run icon.
- The testcases can be run by selectively or at-a-stretch.
- If the selective method is selected, it will ask for the testcase to be run from a list of testcases.
- After selecting the testcase and start running, the silktest will automatically start the application and start the test from the basestate.

**Recording the Events / Action**
Writing scripts in SilkTest includes steps of commands with declaration of window names and its objects before that. To avoid these difficulties and to make the process easier (this is an alternate for writing line-by-line steps of the scripts) silktest provides a special feature of recording events. The steps are given below.

1. Create a new Testcase.
2. Select the option Record – Actions menu.
3. After getting the ‘Record Actions’ dialog box, the sequence of steps to be tested, should be done.
4. ie., the programmer has to simply do the ordinary testing process by selecting or using the windows & its objects.
5. After completing these steps, the user has to click the ‘Paste to Editor’ button in the ‘Record Actions’ dialog box.
6. Now the scripts are automatically available in the script file.
7. Save the script and run that testcase.

A recorded statements for logging in to the yahoo site, will look line the sample given below.

- [-] recording
  - BrowserPage.SetActive()
  - Browser.Location.SetText(www.yahoo.com)
  - Browser.Location.TypeKeys(""")
  - Yahoo.HtmlLink ("Mail|#26|http://www.yahoo.com/r?m2")
  - BrowserPage.HtmlTextField ("Yahoo!ID: | #1"). SetPosition (1,1)
  - BrowserPage.HtmlTextField ("Yahoo!ID: | #1"). Set Text ("username")
  - BrowserPage.HtmlTextField ("Yahoo!ID: | #1"). TypeKeys ("")
  - BrowserPage.HtmlTextField ("Password:|#2"). Set Text ("password")
  - BrowserPage.HtmlPushButton ("Sign In|#1"). Click ()

The alternate for the above recorded statements will be as below:

- Browser.Loadpage (www.yahoo.com) // Loads the yahoo homepage as the default page
- if Yahoo.Exists() // checking for the existence of the homepage
- print ("Yahoo window exists") // confirming that the window exists
- Yahoo.objYahoo.Loginame.SetText ("username")
- Yahoo.objYahoo.Password.SetText ("password")
- Yahoo.objYahoo.Submit.Click()

The difference between the above two different scripts are, the method II needs windows & its objects to declared before the scripts are to be written. It is not in the case of the recording kind of the code.

SILK TEST FEATURES

Platform Independent

Silk Test doesn’t care about how the application is created, in which software the application is written, what kind of design is used, which browser it is being worked, in which operating system the application is running.

All that needs for an application to be tested using silktest is that it needs a frame (like window)

Browser Independent

There are various kinds of browsers used by various people for running their applications. The user may use any browser of his choice to test the standard application. Each and every browser acts differently with different applications. They show the same page differently. The web objects they display can also be aligned or displayed in different manner.

SilkTest just looks at these browser contents as objects and hence they cannot avoid any images, texts,… that they are not identifiable. Also we can write a test in one browser and run it in any other browser (to some extend). i.e…., using SilkTest, we can do cross browser testing.

With minor modifications, your tests are robust enough to support different browsers and different versions of these browsers.
Technology Independent

Silktest does not care how the application was built. It seamlessly works with the different web technologies commonly used today.

How to use the same code for multiple browsers:

Start writing the silk scripts. Capture the window declarations (.inc file) and write the .t file. Say if you are capturing the declarations from Internet Explorer & run successfully on it. As we captured the declarations from I.E., we now have to make the same test case run on Netscape since the tag value changes from multiple browsers.

Testing the Windows based applications
Before start writing scripts, enable the settings given below.

1. Declare all the window names and its objects (used in writing scripts) starting from the first window.
2. In the File-> New option in the menu bar, select the test frame.
3. In the resulting ‘new Test Frame’ dialog box, specify the path of the executable file of your application.
4. After submitting that dialog box, the silktest will automatically create a declaration file with the default window declared.
5. Use that file to create your testscripts.

Testing the Java based applications
Before you start testing the java applications or applets, you have to set the java classpath.

- Point to a Java archive (.jar file) that contains the software the powers SilkTest’s Java support for JDK 1.2 and JRE 1.2. This file is called SilkTest_Java2.jar.
- When you install SilkTest_Java2.jar is installed in this directory:<SilkTest install directory>\JavaEx
If you will use only JDK 1.2 for testing, you can activate Java support for JDK 1.2 by copying SilkTest_Java2.jar from <SilkTest install directory>\JavaEx to <JDK 1.2 install directory>\jre\lib\ext.

If you do not copy SilkTest_Java2.jar to your JDK 1.2 install directory, you must point to it from your CLASSPATH.