

Automated Testing Frame Work

*“Making internet and intranet work
together”*

Ramakrishna Prasad .B
Sunday, November 17, 2002

Introduction

Software development is in a mature stage and this saga requires every software organization to deliver high quality products. A quality product gives the company an edge over all its competitors and can turn business. Organizations across the world have understood this secret and spend in millions of dollars researching on testing methodologies. Numerous people work day in and day out in testing the products. Testing is often considered to be a laborious process requiring many cycles to be run and takes good amount of time in the life cycle of a software development. This led to the need of automation of testing activities. Automation is a one time investment in time and effort. This document describes a frame work in automating test environment for remote systems.

Driving Parameters

The following are the driving forces that lead to the frame work.

+ Accessibility

This is the quality of acquiring any resource when required at any point of time.

+ Open Standards

This will bring in full of advantage of open standards like xml.

+ Platform Independence

The test suites are implemented on remote platforms. This gives the programmer an edge to combine open ended system for proprietary platforms.

+ Usability

Existing tools that comes along with OS can be used. No more extra costs involved.

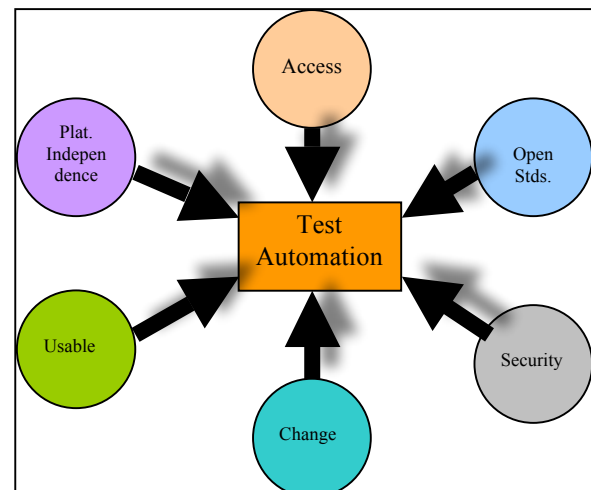
+ Security

The high level of security will be maintained.

+ Changes

The existing test suites need not be re written and can be modified in a least possible manner to comply with this frame work.

Driving Parameters



Functional Requirements

- + The test cases can be run from Remote Client
- + The test cases can be started from
 - o E-Mail (Mobile supporting E-Mail)
 - o A secured server page
- + Frequent updates/ Status will be available through E-Mail / Pager / Secured Server Page
- + Test cases will be stored in repository of remote client and can be transferred during execution
- + Cleans up the host/target system after test execution
- + Logs all activities are maintained during the process
- + A system of Authentication will be introduced for security
- + An system should be made available for users to register in for execution
- + Test summary reports can be generated.

Design

The design of this will involve multiple components/ plug-ins working in a integrated fashion. The design will include the following major components

- + HTTP
- + Mail
- + Repository
- + FTP
- + Telnet
- + Main Application

Components Explained

HTTP

This will be one of the user interfaces to the system. This needs to have two applications developed. One application is for managing the users and the other for selection of tests to execute. The requests will go in as XML. This will send the request to main application and get response from the same.

Mail Server

The mail server component developed will serve as both input and output to the system. The system will accept mail requests and will authenticate the users and will initiate test execution. This will give major flexibility as mails can be sent via a mobile phone. The status can be updated on the Secured Site/ Mobile/Pager. This will send the request to main application and get response from the same.

Repository

This will contain all the test cases. The repository component developed will receive the request from both the sources and will checkout the relevant files. The files can be added to repository manually or a small application can be developed.

Automated Testing Framework

FTP

The FTP server module will download files and information from Repository to remote machine. On completion it will notify the Main module.

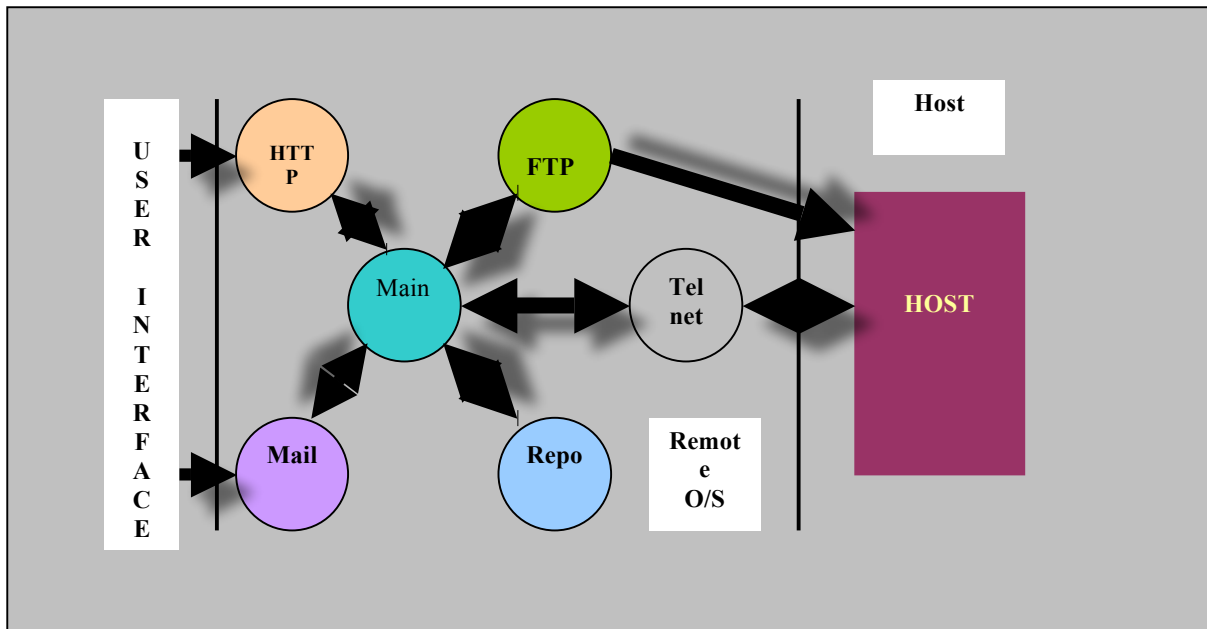
Telnet Server

This is the component that actually executes the test case and sends back the results. This forms the core of the automation. This will be designed to handle mail/xml requests and execute the tests and send back the results in mail/xml formats.

the files to remote system. After successful download it starts the test cases with the help of telnet module and sends back the results.

Conclusion

This framework once developed can also be extended to crucial process that runs for a longer time (Eg.build). This will be extremely helpful and will help productivity be diverted to more useful purposes. I shall post further developments. Please feel free to contact me at ram_prasad_37@yahoo.com



Main Module

This module interacts with the rest of applications. It polls the mail/http module for requests. This process the requests and initiates checkout of files from repository. Once this is done it interacts with ftp modules to download