

# Krypton QA Automation FrameWork

- ✔ No Programing Required
- ✔ Based on Selenium and Spreadsheets
- ✔ Easy to Use
- ✔ Support all Browsers



[DOWNLOAD](#)



Mobile



Browser Testing



Krypton Architecture



Become a Contributor

*Krypton is a Selenium based Open Source framework for your QA Automation needs. It helps Manual QA Testers become experts in Automation without any requirement of programming knowledge. Meant for Web and Mobile it supports common features like Cross Browser compatibility testing, Keyword Driven Testing, Rich Automated Reporting and Parallel Recovery to name some. What makes it different from other freeware's available in market, is its highly customizable property which entertains testing on any kind of project of any*



## Section Summary

<a href="#">Introduction</a>	<a href="#">Quick Start</a>
<a href="#">Supported Platforms</a>	<a href="#">Execution Environment</a>
<a href="#">How to Install Krypton Setup</a>	<a href="#">How to write Test Cases</a>
<a href="#">Testing Process</a>	<a href="#">Parameters Files</a>
<a href="#">Keywords</a>	

## INTRODUCTION

Krypton is a generic framework which has been specifically designed to cater to needs of any client who seeks a low cost end-to-end test automation solution for web using open source tools. Krypton is designed in such a manner that the testers do not need to possess any scripting background at all to automate complex testing scenarios. Everything is already built-in and can be configured easily.

### PRIMARY CHARACTERISTICS OF KRYPTON:

1. Should support multiple web browsers: Internet Explorer, Firefox, Safari, Chrome etc.
2. Should incorporate the 4Rs: Repeat-ability, reliability, re-use and robustness.
3. Should be a user-friendly solution

### KRYPTON IS COMPOSED OF THE FOLLOWING COMPONENTS:

1. Test Engine: is the brain of the framework which will handle all other components
2. Test Driver: will drive automation tools like Selenium integrated with Krypton framework
3. Reporting Engine: will generate reports/log files for automation execution and email notifications.
4. Test Manager: will be responsible to retrieve data from different test management source systems (E.g. File System).

## SUPPORTED PLATFORMS

### 1. Supported Operating Systems

- Windows (XP, Vista, Win 7 , Win 8.1 & Win10)

### 2. Supported Browsers

- Google Chrome 30 or later  
**chromedriver.exe** is already included in setup. After installation, It will be in output folder.
  - Internet Explorer 10.0 or later
  - Mozilla Firefox 19.0 or later (upto 40.0.3)

## HOW TO INSTALL KRYPTON SETUP

### Enough of background study? Let's get started with the product:

Here you will be acquainted with how to quickly configure and run your test cases. Some sample test cases are provided along for easier derstanding with installation package. Everything is pre-configured for the sample test cases.

## LET'S START

Ensure all pre-requisites, download Krypton & run setup file and follow steps.



Click on 'Next'



Click on 'Next'



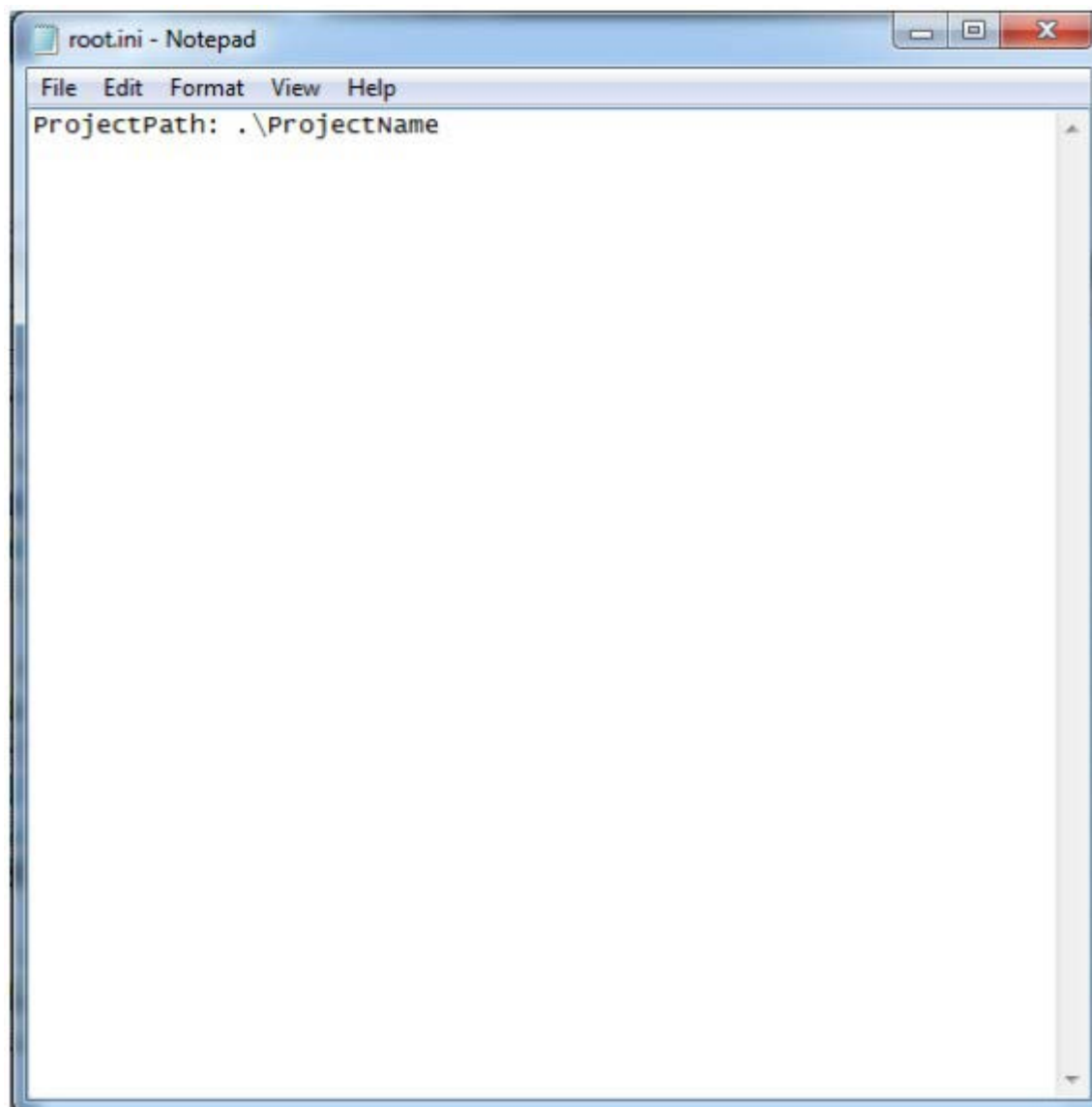
Wait while installation completes. Then the 'Confirm Installation' shows up.



Click on 'Next'



And the installation is complete.



By default Krypton will be installed under C drive.

Sample Test cases and its Object Repository is already provided under TestCases and Object\_Repository folder respectively under Krypton folder.

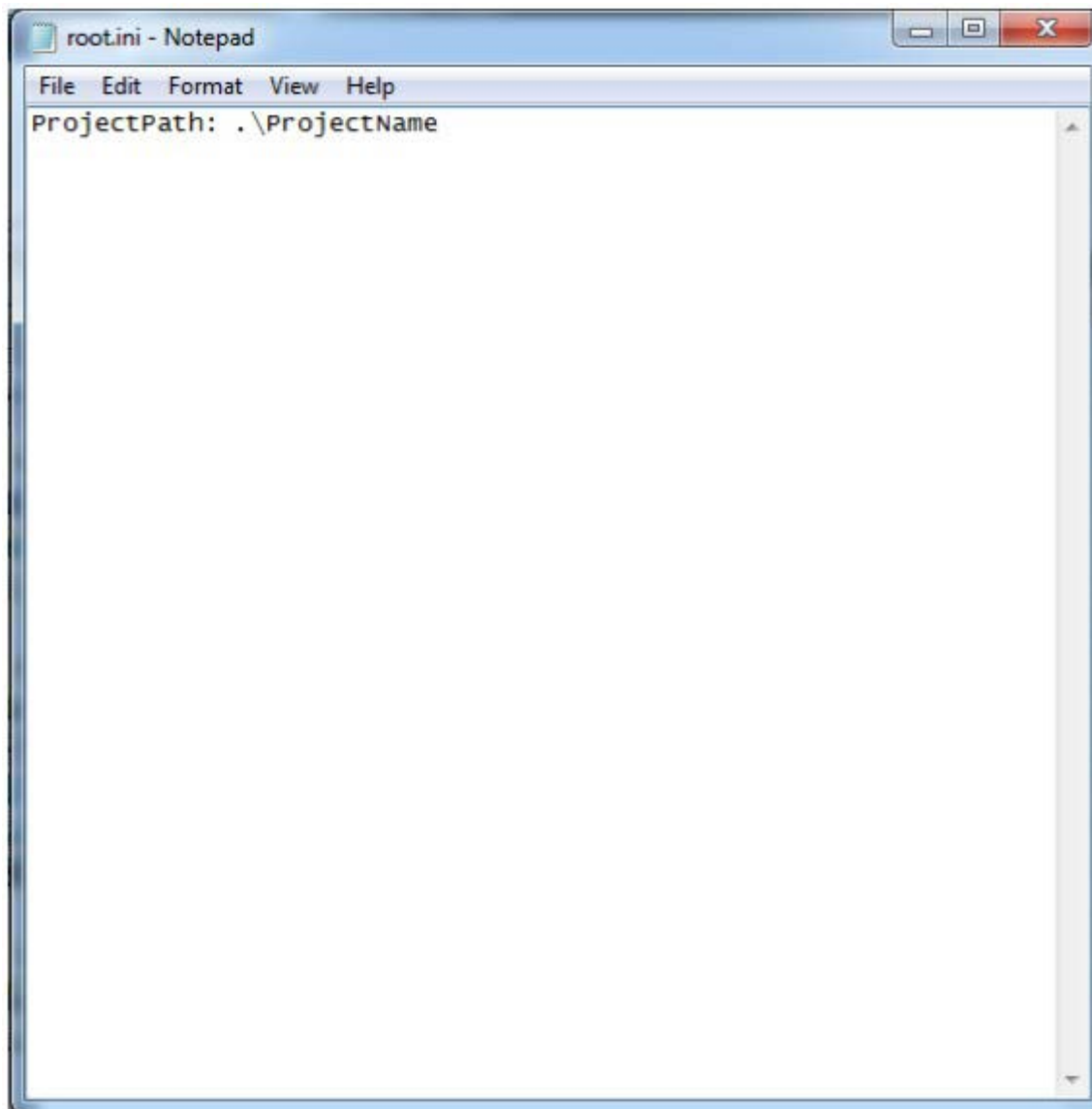
Run the executable file for execution of sample test cases

After execution results will be uploaded at Results folder under respective Project folder.

**START WITH WRITING NEW TEST CASES AND CONFIGURE THE SETTING FOR EXECUTION**

Create new Test case spreadsheet same format as of SampleTest case under TestCases folder. Write reusable test case in Action\_sheet.xlsx file if any in same format placed under TestCases folder.

Identify the objects and create new objects in Obj\_Repository.xlsx file in same format placed at Object\_Repository folder.



Mention test case id and Configure parameters.ini file accordingly and run Krypton.exe







```

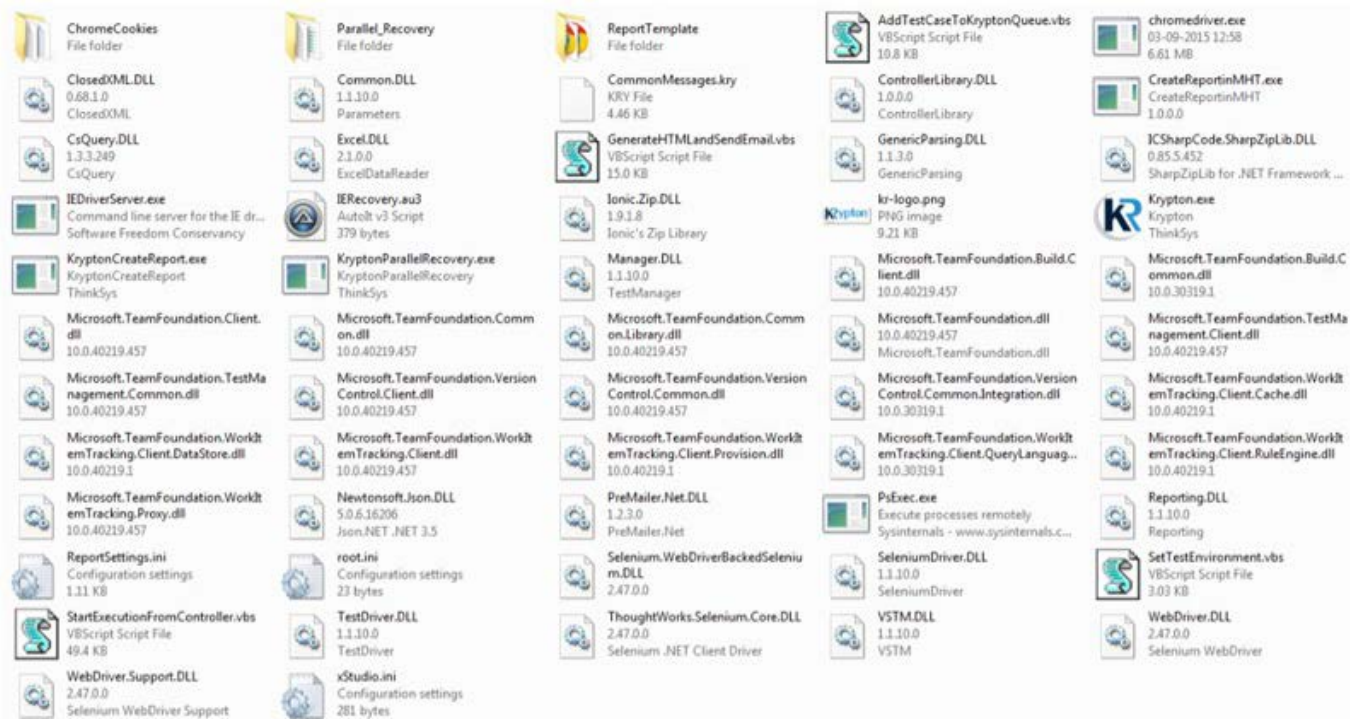
Parameters.ini - Notepad
File Edit Format View Help
Browser: Firefox
SupportedBrowser: IE, Firefox, Chrome, saucelabs
ProjectPath:true
ProjectName:Incredimail
TestSuiteLocation:.\TestSuites
Driver: selenium
DriverSupported: selenium, watir, qtp, testcomplete
TestCaseLocation: .\TestCases
TestDataLocation: .\TestData
TestCaseFileExtension: .xlsx
ReusableLocation: .\Reusables
ReusableDefaultFile:Action_Sheet
ORLocation: .\Object_Repository
ORFileName: Obj_Repository
RecoverFromPopupLocation: .\Parallel_recovery
RecoverFromPopupFileName: RecoveryFromPopups
RecoverFromBrowserLocation: .\Parallel_recovery
RecoverFromBrowserFileName: RecoveryFromBrowser
ParallelRecoverySheetName:popuprecovery
DBTestDataLocation: .\DBTestData
LogFileName: executionlog.xml
LogDestinationFolder: .\Results
TestCaseId:
TestSuite:TC_Demo
ManagerType: FileSystem
SupportedManagerType: FileSystem, QC, XStudio, MSTestManager
TimeFormat: hh:mm:ss tt
SupportedTimeFormat: HH:mm:ss, HH:mm, hh:mm:ss tt, hh:mm tt
DateFormat: MM/dd/yyyy/
SupportedDateFormat: dd/MM/yyyy; MM/dd/yyyy; MMM dd,yyyy
DateTimeFormat: dd/MM/yyyy HH:mm:ss
SupportedDateTimeFormat: yyyy MM dd HH:mm:ss, dd/MM/yyyy HH:mm:ss
CompanyLogo:kr-logo.png
FailedCountForExit:50
SpecificProjectName:
EndExecutionWaitRequired: true
SupportedEndExecutionWaitRequired: true, false
DBConnectionString: DRIVER={SQL Server};server={DbServerName};connection
timeout=6000;APP=TAGS;Trusted_Connection=True;
DefaultDb: write
DBServer: MSSQL
SupportedDBServer: MSSQL, MySQL
DBQueryFilePath: .\SQLQueries
DebugMode: false
ErrorCaptureAs: html
SupportedErrorCapture: image, html
Environment: Env_QA1
EnvironmentFileLocation:.\TestEnvironment
EnvironmentSetupBatch:
TestCaseIDSeperator: .
TestCaseIDParameter: 0
SupportedTestCaseIDParameters: 0,1
SnapshotOption: On failure
SupportedSnapshotOptions: Always, On page change, on failure, Never
RunRemoteExecution: false
SupportedRunRemoteExecutionValues: true, false
RunOnRemoteBrowserUrl: http://ondemand.saucelabs.com
EmailNotification: false
SupportedEmailNotificationValues: true, false
EmailNotificationFrom: krypton.support@thinksys.com
EmailSMTPServer: smtp.gmail.com

```

The project folder gets created outside the Krypton folder with name specified in root.ini file.

 Demo	02-09-2015 17:30	File folder	
 Output	03-09-2015 18:06	File folder	
 Krypton.exe	03-09-2015 17:27	Shortcut	3 KB
 root.ini	03-09-2015 17:27	Shortcut	3 KB

The Krypton folder and its inside looks like this.



## TESTING PROCESS

Testing with KRYPTON involves 4 main stages:

### Creating Tests

1. Now write your test cases in a Excel sheet.
2. Prepare Object repository in a Excel sheet (Obj\_Repository.xlsx).

### Configuring Test:

After writing your test cases and preparing object repository you need to configure your test cases. For configuration follow the steps given:

1. Go to the installation folder of KRYPTON and search for the “parameters.ini” file.
2. Open the file and change following parameters according to your Test:
  - LogDestinationFolder
  - TestCaseld
  - TestSuite

### Running Test:

After configuring you can run your test by just running KRYPTON.exe file or double clicking KRYPTON shortcut on your desktop. You can also set parameters by sending them through command line. E.g. C:>KRYPTON.exe testcaseid=abc.tc01

### Analyzing Test Results:

After test execution completed reports are generated in the **Results** folder. Reports are generated in two forms

1. **Zip form:** All HTML reports are zipped in a file. User can use these reports by unzipping the zipped file.
2. **Unzip form:** All HTML reports are placed in Results folder by the KRYPTON Reporting Engine

## KEYWORDS

Krypton Keyword are words used in TestCases for specific operations. All Keywords used by Krypton are given below.

S.NO	KEYWORD	DESCRIPTION	PARENT	TEST OBJECT	DATA
<b>BROWSER OPERATIONS</b>					
1.	openbrowser	Opens the browser. Without using 'Test Data', it redirects to a blank browser. However, with 'Test Data', it redirects to the URL specified in 'Data Column'			Test Data :- URL (optional)
2.	clearbrowsercache	Clear Browser cache			
3.	closebrowser	Closes running Browser.			
4.	closeallbrowsers	Closes all the running Browsers.			

53. **ObjectTimeout:** Mention waiting time in seconds for an object to appear on web page. If object is not present after the wait time completed, then it raise exception that object not found
54. **GlobalTimeout:** Mention the waiting time for all the actions to be performed before timeout
55. **MaxTimeoutForPageLoad:** It determines the maximum time required for a web page to load
56. **MinTimeoutForPageLoad:** It determines the minimum time required for a web page to load
57. **RecoveryCount:** Mention the number of times recovery can be done
58. **ListOfUniqueCharacters:** "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z"