LCI VALIDATION

LCI VALIDATION APPLICATION

Abstract

This manual will help you to use serialPortValidation application.

Homayoon Ezabadi

Homayoon@simplernetworks.com

Table of Contents

Document version

Author	version	Date	comments
Homayoon Ezabadi	V 0.1	10/30/2015	First draft

Summary:

This application tests the ez-edge serial port interface (LCI). It sends commands read from a yaml file to the NE and verifies its response.

For each sent command, the application create an item in a log file, composed of command sent, expected value, received value, and test result. Log files will be stored at "loggerPath:"

Before running the application

Before running the application, modify all necessary parameters in the "variables.yaml" file. Then make sure all addressed files in variables.yaml file are updated according serial port structure.

####********** variables.yaml **********

```
port : 'COM3'                                   #Port should be the exact name as system port
baudrate : 9600                                 #ez-edge just support 9600
parity : 'N'                                    #ez-edge just support 'N' (serial.PARITY_NONE)
stopbits : 1                                    #ez-edge just support  (serial.STOPBITS_ONE)
bytesize : 8                                    #ez-edge just support  (serial.EIGHTBITS)
timeout : 0.5                                   #the best one is 0.5
serialUser : 'admin'
serialPass : 'ez-edge#1'
#serialPass : 'pass of day'                     #please also change "fserconfStrc" to serialConfigurationTestPassOfDay.yaml
fserconfStrc : 'serialConfigurationTest.yaml'   #this file will use as serial port configuration test references'
#fserconfStrc : 'serialConfigurationTestPassOfDay.yaml'   #this file will use as serial port configuration test references when pass of day is provided'
fStructure : 'serialStructure.yaml'             #load ez-edge LCI structure
loggerPath: 'c:\\logs\\LCI_logs\\'              #all logs will save there
scriptFile: 'scriptFile.yaml'
```

####********** variables.yaml **********

Prepare serial port structure:

To start a test the application needs two yaml files as serial structure:

First one is "fserconfStrc" structure tree of all pages, options and details of configurations. Second one "fStructure" contains Conditions and their prompts.

Structure tree "fserconfStrc":

"fserconfStrc" is structure tree of LCI, yaml structure is like :

```
item1 : value1
item2 : value2
...
```

But values also could be trees of subItems and values:

```
item1 :
    subItem1-1 :
        subItem1-1-1: value1-1-1
            ...
     subItem1-2: value1-2
     ...
item2 : value2
```

In "fserconfStrc" items are main menu options and subItems are options in child pages; "serialConfigurationTest.yaml" version1 in package is designed according current LCI structure (b279), it should be updated after any change in LCI structure.

PageOption

Integer number that shows options in each pages. It should be '0' if page is configuration page and does not have options to proceed to next pages.

ExpectedPage

This should be unique part of page header text, application use this parameter to identify each page of menu.

PageSettings

In this section you should add steps for configuration, for example system configuration needs 3 steps (1- system size configuration 2- Funnel card type 3-confirmation) each steps should have default value and expected prompt and wrong value.

**********"serialConfigurationTest.yaml"**********

```
1 :
 expectedPage : 'setupMenu'
 PageOptions :
  1 :
    expectedPage : 'sysconfig'
    PageOptions : '0'
    PageSettings :
      step1 :
       defaultValue : 1
       expectedPrompt : 'Enter your selection ['
       wrongValue : '3'
      step2 :
       defaultValue : 1
       expectedPrompt : 'Enter your selection ['
       wrongValue : '3'
      step3 :
       'N' : 'EZ-EDGE Local Craft Interface - Configuration Parameters'
       'Y' : 'Changes applied successfully. Press any key to continue'
       expectedPrompt : 'Please confirm you want to proceed with the new configuration [N]:'
       'enter' : 'EZ-EDGE Local Craft Interface - Configuration Parameters'
```

```
      wrongValue : '333.128.3.3'
2 :
  expectedPage : 'netConfig'
  PageOptions : '0'
  PageSettings :
    step1 :
     defaultValue : '192.168.3.120'
     expectedPrompt : 'EZ-EDGE IP address ['
     wrongValue : '333.128.3.3'
    step2 :
     defaultValue : '255.255.255.0'
     expectedPrompt : 'EZ-EDGE Subnet mask ['
     wrongValue : '333.128.3.3'
    step3 :
     defaultValue : '192.168.3.1'
     expectedPrompt : 'EZ-EDGE Default Gateway ['
     wrongValue : '333.128.3.3'
    step4 :
     'N' : 'Configuration Parameters'
     'Y' : 'Changes applied successfully. Press any key to continue'
     expectedPrompt : 'Please confirm you want to proceed with the new configuration [N]:'
     'enter' : 'EZ-EDGE Local Craft Interface - Configuration Parameters'
     wrongValue : '333.128.3.3'
        ...
        ...
```

Conditions and their prompts "fStructure":

The application needs "fStructure" to perform automatic navigation inside LCI, this file is called "serialStructure.yaml" and should be addressed in variables.yaml file.

This file lists conditions and their prompts, each unique prompt allows the program to find where we are exactly in the menu structure.

If a prompt is unrecognized, add a prompt and a returnCode for it at the end and treat it in the program.

For menu page that has return option add return code to "returnCode". For example in "Configuration Parameters" page in LCI, return code is 4 and unique text to identify this page is "EZ-EDGE Local Craft Interface - Configuration Parameters".

For menu page that they don't have return option, for example "System Configuration" does not have return option, put a number between 401 and 499 as return code.

For all not logged in page put a number more than 1001 as return code.

**********"serialStructure.yaml"**********

```
wait:
 idf: 'Too many invalid login'
 returnCode: 1001
notlogin:
 idf: 'NOTICE: This system is strictly'
 returnCode: 1002
badLogin:
 idf: 'Error: incorrect login'
 returnCode: 1003
mainMenu:
```

```
  idf : 'EZ-EDGE Local Craft Interface - Main Menu'
  returnCode : 5
setupMenu :
  idf : 'EZ-EDGE Local Craft Interface - Configuration Parameters'
  returnCode : 4
sysconfig :
  idf : 'EZ-EDGE Local Craft Interface - System Configuration'
  returnCode: 401
sysconfigNoDB :
  idf : 'Enter your selection []:'
  returnCode : 402
```

Script File:

The "scriptFile" file contains a list of commands and their expected result. Application uses this file to do the LCI configuration and validated the output of LCI.

Please note validation is based on response from LCI not checking actual setting. For example via script you could change IP address, validation is based on feedback of LCI not actually test network port.

Script structure:

Script structure should be according below example:

Name of test: testType ; param1 ; param 2 ...
(***Note: this "***Name of test" ***should be unique name***)
Note: for backspace in all scripts put \x08

Supported test types:

serialLoginTest

This script will test 3 different login scenarios, login, bad login and block after 3 bad user/pass (wait). Application will logout LCI automatically if initially LCI is logged in.

Name of test: serialLoginTest ; user ; pass ; (login / badlogin / wait)

autoLogout

This script will log in to LCI (if needed), then it will navigate to given page (first parameter), it could be any page of configTest, then it will wait x seconds (second parameter) then it checked if system logged out automatically.

Name of test: autoLogout ; netConfig (one of configTest pages) ; 901 ; error/pass

configTest

Application supports 6 different type of configurations:

- ***netConfig***
  - This script will change network configuration, please note if you put "y" at the end, system will be reboot automatically, test will resume after boot.
  - **Name of test: configTest ; netConfig ; ip ; mask ; gateway ; y/n**

- ***Sysconfig***
  - This script will change ez-edge system configuration, please note if you put “y” at the end, system will be reboot automatically, test will resume after boot.
  - **Name of test: configTest ; sysconfig ; 1 , 1 . Y/N**
- ***timeConfig***
  - This script will change system time, please note if system has NTP server change time is not possible.
  - **Name of test: configTest ; timeConfig : 2015-10-12 ; 12:10:09 ; Y/N/error**
- ***currentAlarms***
  - This script will navigate to alarm page, validation is based on navigation to alarm page, it could not validate alarm conditions.
  - **Name of test: configTest ; currentAlarms ; enter**
- ***Reset***
  - This script will restart the system, please note if you put “y” at the end, system will be reboot automatically, test will resume after boot.
  - **Name of test: configTest ; reset ; Y/N**
- ***changePass***
  - This script will change system password, you can change pass, test bad current pass, different new pass and confirm pass and bad new pass format.
  - **Name of test: configTest ; changePass ; ez-edge#2 ; ez-edge#1 ; ez-edge#1 ; Y/N/error**

Delay

To perform delay in test use delay : x; when your script needs reboot please use delay more than 15s after it.

Name of test: delay; 15 (name of test should be unique**)**

#####**********'scriptFile.yaml '**********

```
loginSimple1 : serialLoginTest ; admin ; ez-edge#1 ; login
loginSimple2 : serialLoginTest ; adma\x08in ; ez-edge#1 ; login

badLogin1 : serialLoginTest ; adminn ; ez-edge#1; badlogin
badLogin2 : serialLoginTest ; admin ; ez-edge#2; badlogin

badLoginBlock1 : serialLoginTest ; admin ; ez-edge#111 ; wait

alarm1 : configTest ; currentAlarms ; enter


changePassInvalidNew1 : configTest ; changePass ; ez-edge#1 ; ez-edg ; ez-edg ; error
changePassInvalidcurent1 : configTest ; changePass ; eb\x08z-edge#1 ; ez-edge#1 ; ez-edge#1 ; error
changePassDifferentNew1 : configTest ; changePass ; ez-edge#1 ; ezdge#3 ; ezge#2 ; error
changePassEnter1 : configTest ; changePass ; enter ; enter ; enter ; n
changePass1 : configTest ; changePass ; ez-edge#1 ; ez-edge#2 ; ez-edge#2 ; n
changePass2 : configTest ; changePass ; ez-edge#1 ; ez-edge#2 ; ez-edge#2 ; N
changePass3 : configTest ; changePass ; ez-edge#1 ; ez-edge#2 ; ez-edge#2 ; Y

delay1 : delay ; 10

badLogin3 : serialLoginTest ; admin ; ez-edge#1 ; badlogin

loginSimple3 : serialLoginTest ; admin ; ez-edge#2 ; login
```

```
changePass4 : configTest ; changePass ; ez-edge#2 ; ez-edge#1 ; ez-edge#1 ; Y

delay2 : delay ; 10

resetTest1 : configTest ; reset ; N
resetTest2 : configTest ; reset ; Y

delay3 : delay ; 60

configTest4 : configTest ; sysconfig ; 1 ; 2 ; N
configTest4 : configTest ; sysconfig ; 1 ; 23 ; N
configTest5 : configTest ; sysconfig ; 1 ; 1 ; Y

delay4 : delay ; 60


netConfig1 : configTest ; netConfig ; 192.168.3.115 ; 255.255.255.0 ; 192.168.3.1 ; N
netConfig2 : configTest ; netConfig ; 192.168.3.433 ; 255.255.255.0 ; 192.168.3.1 ; N
netConfig3 : configTest ; netConfig ; 192.168.3.115 ; 255.255.430.0 ; 192.168.3.1 ; N
netConfig4 : configTest ; netConfig ; 192.168.3.115 ; 255.255.255.0 ; 192.168.3.1 ; Y

delay5 : delay ; 60

timeConfig1 : configTest ; timeConfig ; 2015-140-12 ; 12:150:09 ; N
#timeConfig2 : configTest ; timeConfig ; 2015-140-12 ; 12:150:09 ; error   #needs ntp server to be available
timeConfig3 : configTest ; timeConfig ; enter ; enter ; N
timeConfig4 : configTest ; timeConfig ; 2015-10-20 ; 14:20:30 ; Y


autologout1 : autoLogout ; netConfig ; 10 ; error
autologout2 : autoLogout ; netConfig ; 901 ; pass
########**********************************************************************
```

Run tests:

After you prepare serial port structure and script file in “variables.yaml”.

Run serialPortValidation.exe and check your log files in “loggerPath” folder. Results will be clearly identified.