Application development to automate platform code download to Set top box

Submitted By

EKTA S. JAYSWAL

16MCEC06

DEPARTMENT OF COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2018

Application development to automate platform code download to Set top box

Major Project

Submitted in fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

EKTA S. JAYSWAL

(16MCEC06.)

Guided By

Prof. Vishal Parikh

DEPARTMENT OF COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

Dec 2017

Certificate

This is to certify that the major project entitled **"Application development to automate platform code download to Set top box"** submitted by **EKTA S. JAYSWAL (16MCEC06)**, towards the fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-II, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Vishal Parikh
Guide & Assistant Professor,
CE / IT Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Priyanka Sharma
Associate Professor,
Coordinator M.Tech - CSE
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr Alka Mahajan
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **EKTA S. JAYSWAL**, **16MCEC06**, give undertaking that the Major Project entitled "**Application development to automate platform code download to Set top box**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Prof. Vishal Parikh

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Vishal Parikh**, Assistant Professor, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- EKTA S. JAYSWAL

16MCEC06

Abstract

Intelligent Automation of manual process has become current trend of modern era. Automation can reduce man effort and time effort drastically. Automation is the process of connecting different or dissimilar systems or software in a manner that they behave as self-regulating entities. Usually this is done so that mundane tasks can be done more efficiently and effectively, thus cutting about time. This application is basically developed for ARRIS internal use only with the agenda of upgrading/downgrading multiple Set-top boxes(STB) automatically through a common procedure. The way we get updates for our devices operating systems, STB would also have up-gradation and down-gradation of platform version and other applications on top of it. Up to now it was a typical manual process consuming so much of human efforts for the simple repetitive tasks. Here my project is to develop a third party plugin which is also a standalone executable which can be imported in AutoMOTO (ARRIS internal tool) to automate the whole code download and version verification process. I have targeted STB's of different category based on the supported protocols for the Code Download process. There are two modules which supports DAC based CDL and TFTP based CDL and automated the whole process. Return On Investment(ROI) report indicates the clear achievement of eliminating hundreds of staff days efforts.

Abbreviations

ASTB	Advanced Set-top Box
CSD	Code Suite download
D5K	DCT5100 and going forward platforms
DAC	Digital Addressable Controller
DCT	Digital Consumer Terminal
DLS	Download Server
DSG	DOCSIS Set-top Gateway
ECDS	Entitlement Control Data Structure
EMM	Entitlement Management Message
F/W	Firmware
LOD	Loader Object Directory
MSO	Multiple System Operator
OOB	Out-of-band
PROS	Permissions Resource Object Signatory
RADD	Remote Addressable DANIS/DLS 6000
RAM	Random Access Memory
SDG	Software Download Generator
CDL	Code Download
STB	Set-top Box
TC	ThinClient Object

Contents

List of Figures

Chapter 1

Introduction

Automation is the process of connecting different or dissimilar systems or software in a manner that they behave as self-regulating entities. Usually this is done so that mundane tasks can be done more efficiently and effectively, thus cutting about time. Development and Test team would identify why the specific issue was introduced and identify the action items to prevent it in future. Television Now-a-Days has become a necessity rather than a source of joy, and Set top boxes are the heart of these television. STB would also have F/W up-gradation and down-gradation, the manual process is automated in this project. ELM STBs and ThinClient STBs are targeted for CDL automation.

1.1 Code Download a Manual Process

The way we get software updates in our mobile devices, Set-top boxes also would have firmware version upgradation or down gradation as per requirement. A manual process includes: Either using USB flash method or manual TFTP based process, which consumes so much of time if need to handled for many boxes at the same time. Also some error and failing scenarios need to be handled, debuggable and report the possible reasons. So there is a need to Automate Code download procedure.

1.2 Project objective and benefits

Develop a plugin which can work with an internal framework called AutoMoto(ARRIS Internal Tool) to automate code download procedure for set top boxes. Different categories of STBs need to be targeted based on the supported protocols for Code Download process such as DAC, TFTP etc and automated the whole process of firmware/application

up-gradation and down-gradation.

Benefits: Suppose a RAC is having 100 STBs connected and all boxes need to be up-graded then earlier it used to take 1 Day Man efforts. But after this plugin in use, it can reduce it to some minutes procedure only. In manual process if something goes wrong its difficult to debug and report. But this plugin will handle all possible error cases and will take actions accordingly and can report all the execution details along with possible problems.

1.3 Block Diagram for adding extra functionality by 3rd party Plugins(medium)

This block diagram indicates that to add any new additional functionality you can create a plugin in .net and generate a class library file ".dll" file. Then by adding the plugin into AutoMoto medium manager it can be used according to requirement.

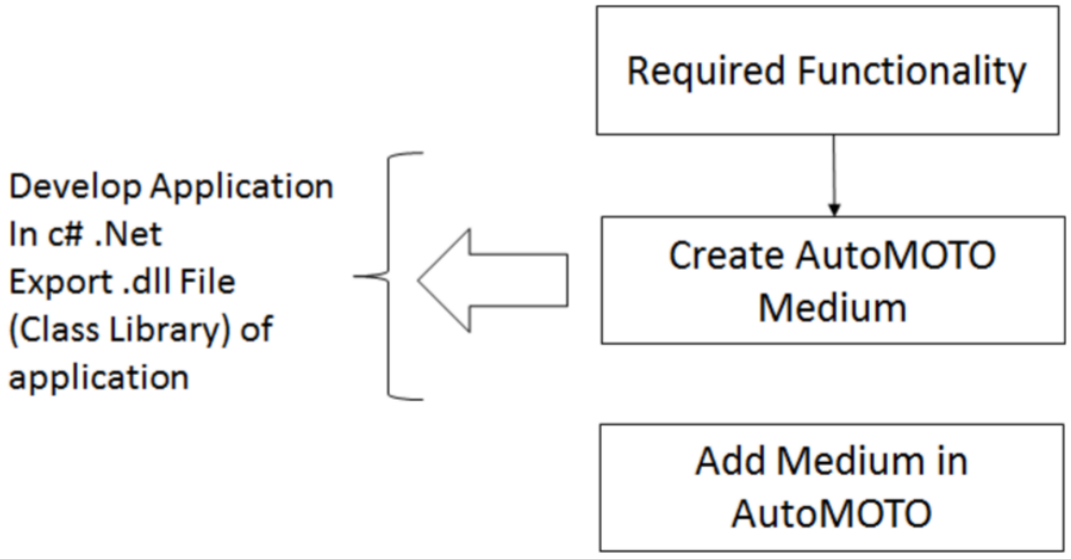


Figure 1.1: Block Diagram for adding extra functionality by 3rd party Plugins(medium)

Chapter 2

Tools and Technologies used

2.0.1 Visual Studio 2015 (C#.net)

Microsoft Visual Studio is used as development environment of our plugin. C#.net pogramming language is used for the development.

2.0.2 AutoMOTO (ARRIS Testing Tool)

Version: 20.19 AutoMOTO is a framework, developed by ARRIS developers for their devices only.

It can be used by developers as well as testers to automate test procedures, that otherwise would have consumed a lot of time.

Using AutoMOTO, a tester can create, manage,schedule, and review the results of automated tests.

The medium used is a .Net DLL program object which helps in communication between AutoMOTO software and various testable devices via a software or hardware interface (such as COM port, USB devices, TCP/IP and SNMP protocols etc.).[1]

2.0.3 WINSCP

Win SCP(Windows Secure Copy) is a free and open source SFTP, FTP, WebDAV and SCP client for Microsoft Windows. Its main function is secure file transfer between a local and a remote computer.[2]

2.0.4 Tera Term

Tera Term is an open-source, free, software implemented, terminal (communications) program. Which is used to interact with the serially connected device with computer. This tool allows to configure each port with typical custom configurations.[3]

2.0.5 Selenium (Webdriver, configuration, version, script)

Version: 3.5 latest Selenium web driver Selenium is a suite of testing automation tools used for Web-Base applications: Selenium IDE, Selenium RC, Selenium WebDriver and Selenium Grid. These tools provide a rich set of testing functions specifically geared to varied testing scenarios of all types of Web applications. The operations provided by these tools are highly flexible and afford many options for comparing UI elements to expected application behavior. Selenium tests can be executed on multiple browser platforms.[4]

2.0.6 Mozilla Firefox Browser

Version : v56 (Any version ¿55 will be compatible) Mozilla firefox is used along with Selenium automation tool to execute the script for DAC CDL. Headless browser is used so that whole process can be executed in background. Make sure, no firefox instance is opened before starting the execution.[5]

2.0.7 Virtual Serial Port Emulator

VSPE is used to monitor the on going process on port concurrently along with Auto-MOTO script. Its splitter functionality creates a virtual port and maps it to actual physical port with appropriate compatible configurations. This tool enables accessing the same port concurrently, it allows to interact and view the received and sent data on port from our machine.

2.0.8 Set Top Box

A STB is a device that converts a digital television signal to analogue for viewing on a conventional set, or that enables cable or satellite television to be viewed.The signal sources for the set top box might be an Ethernet cable, a satellite dish, a coaxial cable, a telephone line , broadband over power lines,or even an ordinary VHF or UHF antenna. Content, in this context, could mean any or all of video, audio, Internet web pages, interactive video games, or other possibilities. Set top boxes are devices which helps in

displaying the transmitted content onto the TV. [6] Set top box needs 2 basic software elements to work: Firmware and Tv guide.

STB are of different categories in ARRIS:

ELM,ThinClient, RDK, VTR, IRVINE, DELMUR Different categories boxes support different type of protocols for Code Download Procedure.

For DAC based CDL ThinClient STBs and TFTP based CDL ELM based STBs are used.[6]

Chapter 3

System Analysis

3.1 System scope

Test Script developer can use this plugin to develop the test script when code download process initiation and verification is required.

Tester can use it for testing the stability of STB

Tester and test script developer can provide custom runtime parameter as well during runtime.

On site this can be used to upgrade or downgrade STB platform version as well as application objects versions.

3.2 Sytem Features

1. Code download process is supported for Thinclient loader which uses DAC method and KACodeLoader which uses TFTP method.

2. User interface is provided for configuring multiple set top boxes with code loader interfaces, which allows DAC, FTP and serial port (COM Port) configurations.

3. DAC base CDL: Dac Configurations customization for each interfaces is supported.

4. FTP support is provided for automatic upload of build files on server. FTP Configurations customization for each interface.

5. Supports for Platform Object creation, usage as well as uploading using FTP protocol.

6. Supports for multiple Application Object creation, usage as well as uploading using FTP protocol.

7. FTP functionality which can eliminate use of any tool to upload build files on server in TFTP and DAC based CDL both.

8. Automatic deletion of all created objects by script from DAC once CDL process is completed.

9. Supports separate insertion rate for Platform object build as well as Application Object build.

10. Uni-cast, multi-cast and broadcast is supported. In user interface Terminal index and Broad cast both addressing types are supported to accept user inputs.

11. Run time variable support is provided partially.

12. TFTP based CDL: User interface takes input from source, destination, other FTP configurations, platform type and reload checkbox.

13. FTP file upload on runtime as well as Runtime variable support is provided fully for it.

3.3 User Characteristics

User must be familiar with AutoMoto, and should be aware of working of Automoto, creating test steps, configuration, execution, scripting, runtime variable usage etc.

User with less programming skills will be fine.

User should be familier with the set-top boxes.

3.4 Operating environment

This plugin will be used with AUTOMoto, and hence operating environment will be machine having Windows 7 and above operating systems only.

Chapter 4

DAC Based Code Download

4.1 ThinClient Code Download Introduction

The Code Suite Downloader is a normal task like most others in the Thin Client platform. Its initialization performs normal startup functions such as allocating resources needed for continuous processing of download commands from the head-end and API requests from the application and from other parts of the system. There is no destructor since the system is not designed to run without or to replace the Code Suite Downloader.

Part of initialization is to identify and check the integrity of important permanent structures (those maintained across resets), notably the Loader Object Directory. If invalid, it must be recreated. If valid, then all objects in the table are also checked for validity. Objects in transient states (such as to-be-deleted, loading) are deleted. Objects that appear to be using resources that do not exist are marked so that they cannot be enabled.

Once initialized, the Code Suite Downloader waits for messages and API requests. The processing of head-end messages may initiate a long series of operations. API requests may be simple (executed in the context of the caller and completed immediately; e.g. function call), synchronous (queued for execution with response delivered later, and the calling thread is blocked) or asynchronous (queued for execution with response delivered later, calling thread is not blocked). At present, only simple API functions are defined.

When a download is initiated by a head-end ASTB tune download command, the Code Suite Downloader checks the availability of the authentication structure (ECDS) for all signed objects in the list, checks for sufficient resources to store all objects, makes the necessary reservations, and downloads the objects as directed. Since the auto-enable flag

is set as default, all objects in the list are automatically enabled after they are successfully downloaded.

In Code Suite mode, after application objects are loaded, Code Suite relocatable module will handle loading objects relocatably and readers should read reference 12 for details.

If there is support for download path (DSG tunnel or QPSK) determination by the DSG module, then the downloader will wait for a download path to be set by the DSG module before downloading any objects.[6]

Notes: ECDS in this document applies to Non-DCH platforms only. For DCH platforms, PKCS authentication replaces ECDS and PKCS signatures are required for Platform Object and Application Object. For PKCS signature and authentication, the reader should refer to reference 13 for details.[6]

4.2 Flow Diagram :

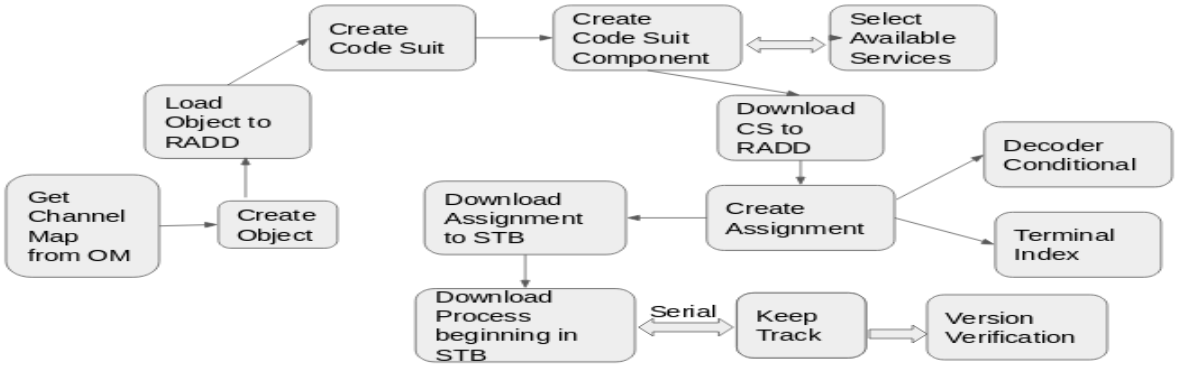


Figure 4.1: DAC Flow diagram

Description:

1 Code Download Process : using Selenium Web Automation tool, create and download the code suit to appropriate STB or STBs.

2 Establish Serial communication and capture the logs from STBs.

3 Analyze logs and Track Code Download Process.

4 Version Verification: Once Download is completed Validate the Platform build version and application object build versions

4.3 Code Download Process

Build a service to support a download

Load objects onto the DAC and then download them to the RADD

Define code suites

Build software object assignments

Monitor the download process

4.3.1 Downloading

The downloading function allows an MSO to deliver, upgrade/update firmware and applications to set-tops remotely or locally

4.3.2 Components of a Download Data Stream

Download Control Messages

Object Modules

4.3.3 Methods of Targeting multiple STBs for Downloads

Decoder Conditional Specific sets of terminals may be addressed through the use of an optional decoder conditional. It is a preamble that contains an expression consisting of decoder conditional terms and logical operators. A decoder conditional may be used in conjunction with any other addressing mode (i.e. broadcast, terminal ID)
When building a decoder conditional on the DAC 6000, two things are entered, they are: Decoder Conditional Name and Expression

4.3.4 Method of Targeting Terminals for Downloads

OM 1000 Configure the Background Service for the OM 1000 that services the Downstream Plant that the set-tops are on

Virtual Channel Map Only add the Background Service to the VCM that is assigned to the set-tops to be downloaded

4.3.5 Object Module - Types

Base Platform/Firmware Applications Monolithic Code Suite

4.3.6 Object Assignment Options

The final steps when downloading are: Define the object assignment on the DAC Load the object assignment to the RADD

4.3.7 Object Assignment Options

Which object assignment to use will depend upon: Set-top type Object type

4.4 Hardware used and their funtions

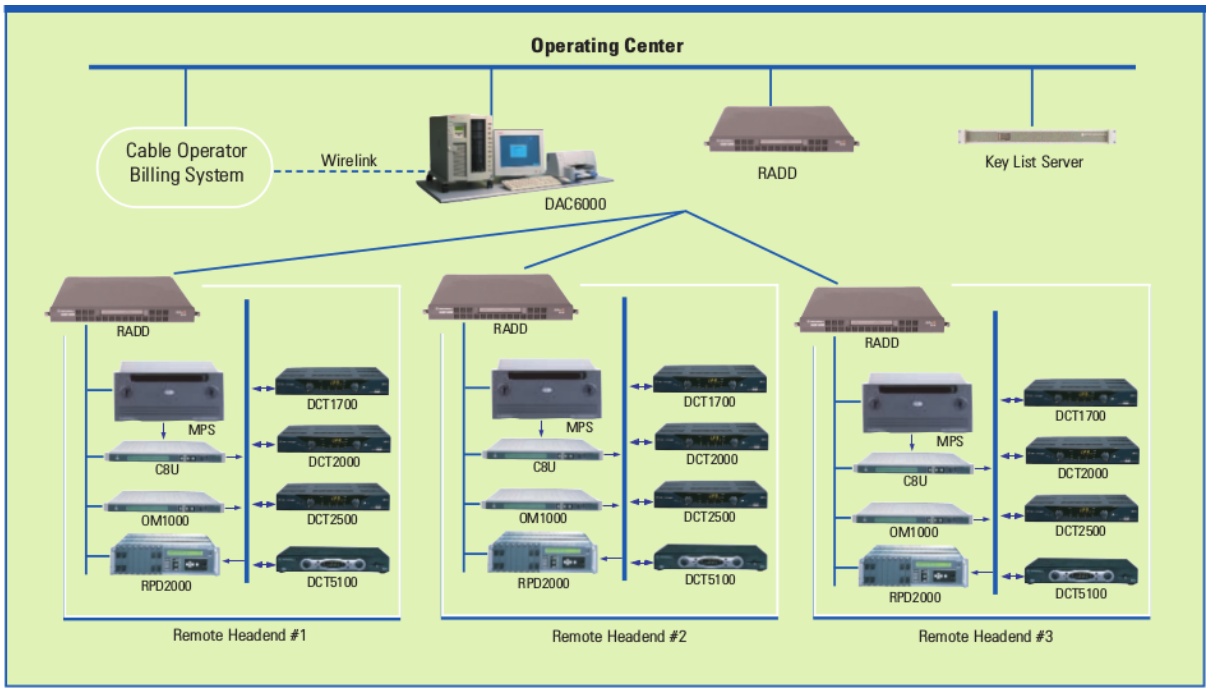


Figure 4.2: DAC based Architecture

4.4.1 DAC6000/RADD6000 Functions

Authorizes video and audio services and operating features for DCTs. Facilitates secure delivery of services. Provides an interface to business systems for provisioning and inventory management. Maintains a database of terminal and headend equipment operating parameters. Manages service databases and automated scheduling of services, including subscriptions, IPPV Collects impulse transactions from terminals and transmits them to the business system. Monitors the terminal population. Supports virtual channels in the digital multiplex.

4.4.2 OM 1000 Functions

Control link through which the digital headend transmits addressable commands and authorization messages that control DCTs. Other types of information delivered OOB: Interactive Program Guide (IPG) or Electronic Program Guide (EPG) data Application code downloads Control messages for emergency situations Figure3.3.2 OM 1000 Executable code downloads to upgrade cable terminal operations Interactive downstream control messages OOB (Out-of-Band) Traffic includes Address Control (Change Services) & Background Services:

1 Download Firmware & Software

2 Send Guide Data (EPG/IPG)

3 Interactive Control (VOD)

4 Handle EAS (Emergency Alert System)

Input Multiple DAC/RADD EMM Digital Streams Output One RF and QPSK Stream In-Band vs. Out-of-Band (OOB) EPG Electronic Program Guide IPG - Interactive Program Guide

4.5 Setup and run

4.5.1 Add medium in Automoto

- In order to use the third party plugin in AUTOMoto CodeLoadMedium.dll file of plugin need to be added in Medium Manager. Please follow below steps for doing the same.
  Click on → Tools → Medium Manager → Add new Medium → Open path of CodeDownload.dll file and select it → Click on Ok

4.5.2 Configure device with Code Loader interface

- Code download plugin need to access set-top box via serial communication. So serial connection need to be configured in order to bind the device with the interface of the plugin. Please follow below steps for doing the same.
  Click on → Edit → Edit Configuration Wizard → Configuration Wizard will be opened.

- Now here we need to create a new device, which we will have to bind with the free interface of Code loader, by dragging and dropping that device on any of available interface of our plugin on right side panel. This will open a user interface of our Code Loaders configuration user interface.
  Select → ThinClient Loader → Select DAC method → This will set up a UI on right side. Here we need to enter appropriate configuration details, which will be used while running test case.

1 **Serial Configuration:** Here Serial Connection parameters need to be selected appropriately such as, COM Port : COM Port with which the STB is connected using serial cable.

- Baud Rate: Enter baud rate compatible with STB. In case of ThinClient box, 38400 should be selected.

- Parity: Odd should be selected for ThinClient box. Other parameters are fine as default one.

2 **DAC Configurations:**
  Here the configurations of the DAC server to be used need to be entered appropriately.
  Note: Default configurations are fetched from the config.xml file. The updated values will be saved in secondary variables of AUTOMoto.

3 **FTP Configurations:**
  Here the configurations of the FTP server to be used need to be entered appropriately.
  Note: Default configurations are fetched from the config.xml file. The updated values will be saved in secondary variables of AUTOMoto.

- Verify all entered data and press ok button. If it was successful then a prompt will be shown along with interface number saying that it got configured and saved successfully.

4.5.3 Create test case for DAC based CDL

- In order to create a new test case for the Code downloader follow the process below. Click on → Create New Test Case → select Utilities → Select Code loader → Drag and drop Icon

4.5.4 FTP build file upload and run test case

- Verify/update FTP server configurations where you want to upload the build files. Click on → Browse Button → Open Build File → Click on Ok

- This will show prompt saying FTP upload was successful or not.
  Follow this process for Application Objects as well.

- Run the test case, make sure to select the device which you configured when asked.

4.5.5 Troubleshooting Tips

Error code and their description:
DL01 - Your object is incompatible with the settop your are targeting
DL02 - The download object can not be found
DL03 - The download object differs in name or size from the object name or size contained in segment 1
DL04 - The download service you have selected is not in your map
DL05 - The object you are downloading is an older version than the Boot version contained in your settop
DL06 - Checksum failure
DL07 - Insufficient storage space (most commonly seen when trying to download platform code before an application has had time to disable and erase)

4.5.6 Version Verification

After certain interval of time send command 1016 this will fetch following logs. This indicates current versions on STB are v34.45 Platform and TvGuide v79.85.

Figure 4.3: Current versions on STB v34.45-Platform and TvGuide-v79.85

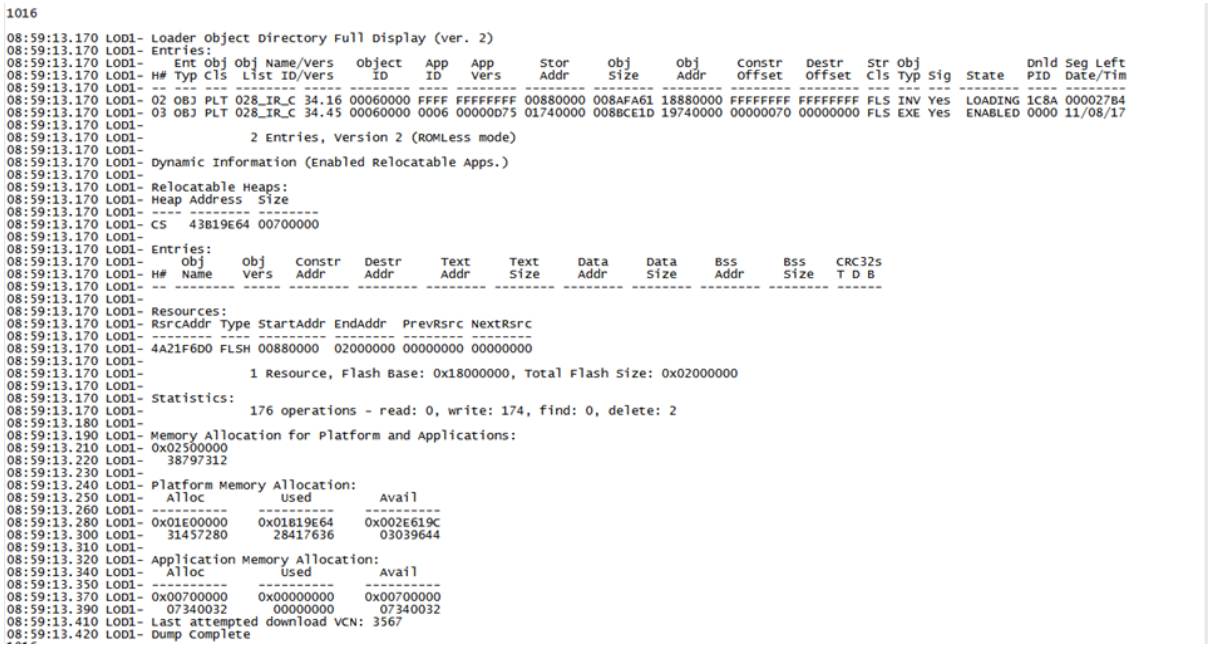
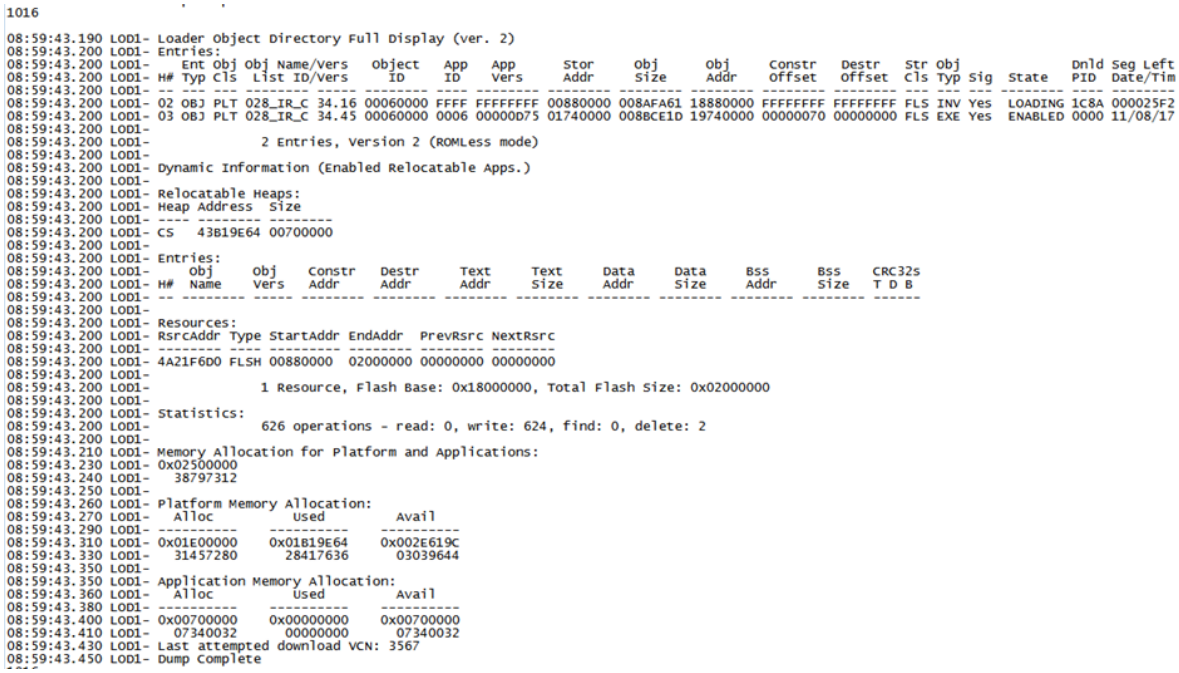


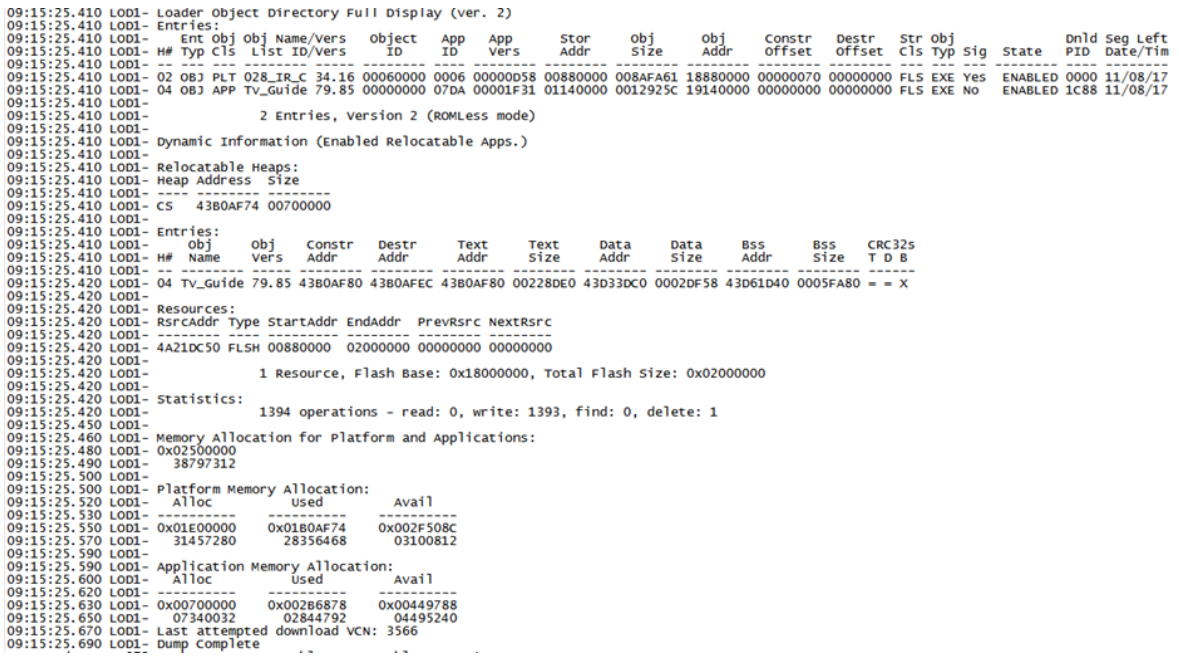
Figure 4.4: Current Version is 34.45 and its being downgraded to 34.16:



```
1016

08:59:43.190 LOD1- Loader Object Directory Full Display (ver. 2)
08:59:43.200 LOD1- Entries:
08:59:43.200 LOD1-    Ent Obj Obj Name/Vers   Object   App   App      Stor     Obj      Obj     Constr   Destr   Str Obj              Dnld Seg Left
08:59:43.200 LOD1- H# Typ Cls  List ID/Vers     ID     ID    Vers     Addr     Size     Addr    Offset   Offset  Cls Typ Sig  State   PID  Date/Tim
08:59:43.200 LOD1- -- --- --- -------- ----- -------- ---- -------- -------- -------- -------- -------- --- --- --- -------- ---- --------
08:59:43.200 LOD1- 02 OBJ PLT 028_IR_C 34.16 00060000 FFFF FFFFFFFF 00880000 008AFA61 18880000 FFFFFFFF FFFFFFFF FLS INV Yes  LOADING 1C8A 000025F2
08:59:43.200 LOD1- 03 OBJ PLT 028_IR_C 34.45 00060000 0006 00000D75 01740000 008BCE1D 19740000 00000070 00000000 FLS EXE Yes  ENABLED 0000 11/08/17
08:59:43.200 LOD1-
08:59:43.200 LOD1-              2 Entries, Version 2 (ROMLess mode)
08:59:43.200 LOD1-
08:59:43.200 LOD1- Dynamic Information (Enabled Relocatable Apps.)
08:59:43.200 LOD1-
08:59:43.200 LOD1- Relocatable Heaps:
08:59:43.200 LOD1- Heap Address  Size
08:59:43.200 LOD1- ---- -------- --------
08:59:43.200 LOD1- CS   43B19E64 00700000
08:59:43.200 LOD1-
08:59:43.200 LOD1- Entries:
08:59:43.200 LOD1-    Obj     Obj    Constr   Destr     Text     Text     Data     Data     Bss      Bss    CRC32s
08:59:43.200 LOD1- H#  Name    Vers   Addr     Addr      Addr     Size     Addr     Size     Addr     Size   T D B
08:59:43.200 LOD1- -- -------- ----- -------- -------- -------- -------- -------- -------- -------- -------- ------
08:59:43.200 LOD1-
08:59:43.200 LOD1- Resources:
08:59:43.200 LOD1- RsrcAddr Type StartAddr EndAddr  PrevRsrc NextRsrc
08:59:43.200 LOD1- -------- ---- --------- -------- -------- --------
08:59:43.200 LOD1- 4A21F6D0 FLSH 00880000  02000000 00000000 00000000
08:59:43.200 LOD1-
08:59:43.200 LOD1-              1 Resource, Flash Base: 0x18000000, Total Flash Size: 0x02000000
08:59:43.200 LOD1-
08:59:43.200 LOD1- Statistics:
08:59:43.200 LOD1-              626 operations - read: 0, write: 624, find: 0, delete: 2
08:59:43.200 LOD1-
08:59:43.210 LOD1- Memory Allocation for Platform and Applications:
08:59:43.230 LOD1- 0x02500000
08:59:43.240 LOD1-   38797312
08:59:43.250 LOD1-
08:59:43.260 LOD1- Platform Memory Allocation:
08:59:43.270 LOD1-   Alloc          Used         Avail
08:59:43.290 LOD1- ----------    ----------    ----------
08:59:43.310 LOD1- 0x01E00000    0x01819E64    0x002E619C
08:59:43.330 LOD1-   31457280      28417636      03039644
08:59:43.350 LOD1-
08:59:43.350 LOD1- Application Memory Allocation:
08:59:43.360 LOD1-   Alloc          Used         Avail
08:59:43.380 LOD1- ----------    ----------    ----------
08:59:43.400 LOD1- 0x00700000    0x00000000    0x00700000
08:59:43.410 LOD1-   07340032      00000000      07340032
08:59:43.430 LOD1- Last attempted download VCN: 3567
08:59:43.450 LOD1- Dump Complete
```

Figure 4.5: 34.16 is Loading



```
09:15:25.410 LOD1- Loader Object Directory Full Display (ver. 2)
09:15:25.410 LOD1- Entries:
09:15:25.410 LOD1-    Ent Obj Obj Name/Vers   Object   App   App      Stor     Obj      Obj     Constr   Destr   Str Obj              Dnld Seg Left
09:15:25.410 LOD1- H# Typ Cls  List ID/Vers     ID     ID    Vers     Addr     Size     Addr    Offset   Offset  Cls Typ Sig  State   PID  Date/Tim
09:15:25.410 LOD1- -- --- --- -------- ----- -------- ---- -------- -------- -------- -------- -------- --- --- --- -------- ---- --------
09:15:25.410 LOD1- 02 OBJ PLT 028_IR_C 34.16 00060000 0006 00000D58 00880000 008AFA61 18880000 00000070 00000000 FLS EXE Yes  ENABLED 0000 11/08/17
09:15:25.410 LOD1- 04 OBJ APP Tv_Guide 79.85 00000000 07DA 00001F31 01140000 0012925C 19140000 00000000 00000000 FLS EXE No   ENABLED 1C88 11/08/17
09:15:25.410 LOD1-
09:15:25.410 LOD1-              2 Entries, Version 2 (ROMLess mode)
09:15:25.410 LOD1-
09:15:25.410 LOD1- Dynamic Information (Enabled Relocatable Apps.)
09:15:25.410 LOD1-
09:15:25.410 LOD1- Relocatable Heaps:
09:15:25.410 LOD1- Heap Address  Size
09:15:25.410 LOD1- ---- -------- --------
09:15:25.410 LOD1- CS   43B0AF74 00700000
09:15:25.410 LOD1-
09:15:25.410 LOD1- Entries:
09:15:25.410 LOD1-    Obj     Obj    Constr   Destr     Text     Text     Data     Data     Bss      Bss    CRC32s
09:15:25.410 LOD1- H#  Name    Vers   Addr     Addr      Addr     Size     Addr     Size     Addr     Size   T D B
09:15:25.410 LOD1- -- -------- ----- -------- -------- -------- -------- -------- -------- -------- -------- ------
09:15:25.420 LOD1- 04 Tv_Guide 79.85 43B0AF80 43B0AFEC 43B0AF80 00228DE0 43D33DC0 0002DF58 43D61D40 0005FA80 = = X
09:15:25.420 LOD1-
09:15:25.420 LOD1- Resources:
09:15:25.420 LOD1- RsrcAddr Type StartAddr EndAddr  PrevRsrc NextRsrc
09:15:25.420 LOD1- -------- ---- --------- -------- -------- --------
09:15:25.420 LOD1- 4A21DC50 FLSH 00880000  02000000 00000000 00000000
09:15:25.420 LOD1-
09:15:25.420 LOD1-              1 Resource, Flash Base: 0x18000000, Total Flash Size: 0x02000000
09:15:25.420 LOD1-
09:15:25.420 LOD1- Statistics:
09:15:25.420 LOD1-              1394 operations - read: 0, write: 1393, find: 0, delete: 1
09:15:25.450 LOD1-
09:15:25.460 LOD1- Memory Allocation for Platform and Applications:
09:15:25.480 LOD1- 0x02500000
09:15:25.490 LOD1-   38797312
09:15:25.500 LOD1-
09:15:25.500 LOD1- Platform Memory Allocation:
09:15:25.520 LOD1-   Alloc          Used         Avail
09:15:25.530 LOD1- ----------    ----------    ----------
09:15:25.550 LOD1- 0x01E00000    0x01B0AF74    0x002F508C
09:15:25.570 LOD1-   31457280      28356468      03100812
09:15:25.590 LOD1-
09:15:25.590 LOD1- Application Memory Allocation:
09:15:25.600 LOD1-   Alloc          Used         Avail
09:15:25.620 LOD1- ----------    ----------    ----------
09:15:25.630 LOD1- 0x00700000    0x00286878    0x00449788
09:15:25.650 LOD1-   07340032      02844792      04495240
09:15:25.670 LOD1- Last attempted download VCN: 3566
09:15:25.690 LOD1- Dump Complete
```

Figure 4.6: New version : 34.16-Enabled

- For Version verification we need to analyze log file of size Kbs or Mbs for a single STBs logs, when we are having hundreds of box it increases the time complexity of the task. Here there are only 5 attributes affecting the final success or failure of CDL process but when we are having thousands of attributes then rather than using a conventional approach, we can apply some intelligence using Machine Learning

approach to analyze such huge logs. I implemented a prototype to demonstrate the power of Machine Learning and how we can apply it in such huge log analysis.

- This process of identifying on whether an upgrade(or a process) has failed or it has succeeded involves to many conditions and is cumbersome .
- Methods of these types on very large LOG data takes hours and sometimes days to run.
- To make it more efficient I proposed an alternate interpretation of the problem , it could be seen as : **Classifying Success or Failure of a process given its other surrounding parameters.**
- This now is a standard machine learning classification problem . And I was given an opportunity to present a prototype of this machine learning approach .
- I accomplished the classification task using a artificial neural network and it was received with great appreciation in the company.
- My approach: Artificial Neural Network
  150 Neurons for each entry in file: 150 rows
  2 Hidden layers: with 256 neurons
  Output: SoftMax → Success/Failure

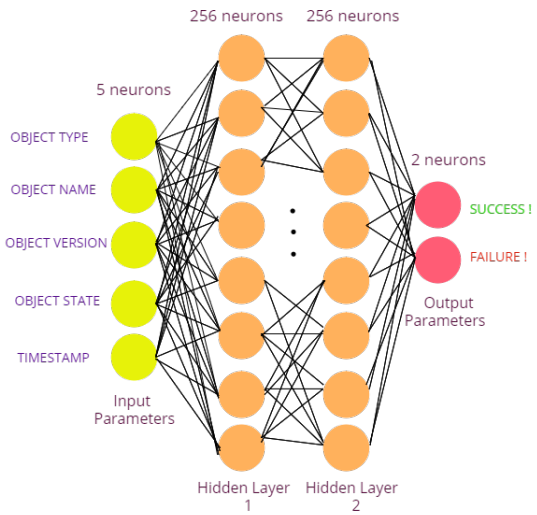


Figure 4.7: The architecture of the ANN

Chapter 5

TFTP based Code Download for Set-top boxes

Category of set-top boxes which supports TFTP protocol, are targeted by this particular code download method of our plugin. Code download initiation as well as verification is will be performed automatically.

5.1 Flow Diagram

A computer, through its Ethernet TFTP port, is a fast and easily obtainable instrument for loading platform code into a set-top box.

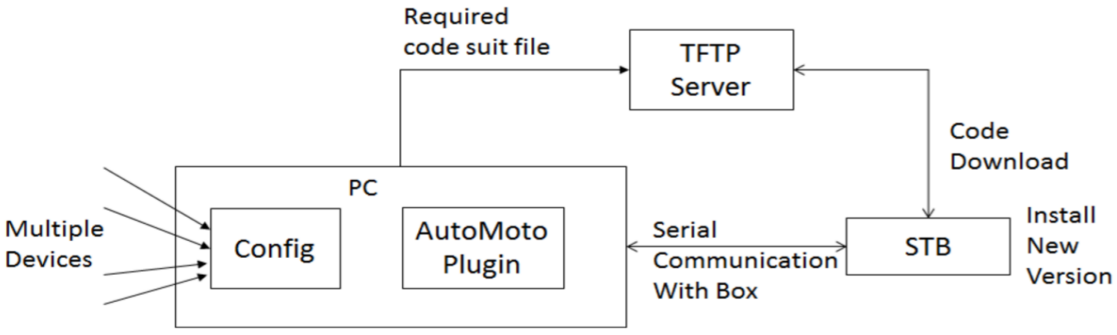


Figure 5.1: TFTP Based CDL: Block Diagram

- Description of the flow:

  - Appropriate Code Suite build must already be kept in TFTP Server

  - Establish Serial Communication from our computer to STB and via this connection perform following tasks.

- Get Current Version on STB via Serial communication
- Reboot STB
- Enter in SSBL boot mode via serial port
- Enter into debug mode via Serial communication
- Run install command with along with TFTP servers Ip and file path
- Track the process by monitoring Serial port data
- After the download completes, the STB will reboot.
- Verify that the expected platform version is running in the serial port output.

5.2 Setup and Run Test Case

5.2.1 Steps

1. To Create Test Step of Code Downloader drag and drop Code Loader from Modules under Utility Category.

2. Enter appropriate information in User Interface of Code Loader.

- Select KALoader as loader and TFTP method

- Enter values in all the fields as follows with $varName. These values can be set at runtime.

- Select Platform Type as ELM.

- Check or uncheck the Reload box. In case if your boxs current version is similar as new version then reload will again load the same build. While unchecked reload will come out with success saying Already Same Build.

- Click on Save.

3. Run the test. Select the device which is configured with Code Loader interface from Config Wizrd. Here make sure you have configured with compatible configurations of the box (e.g baud rate: 115200, parity: none etc).

4. Set Runtime variable values appropriately as follows:
   $host: 10.237.155.19
   $port: 22

$source: (Add path of your local machine)

C:-bi-elmINTERNAL-USE-ONLYKA-trunk.613583bcm45dcx4220.bin

$destination: "/extra/tftpboot/"

$user name: root

$password: root123

5.2.2 Add medium in Automoto

- In order to use the third party plugin in AUTOMoto CodeLoadMedium.dll file of plugin need to be added in Medium Manager. Please follow below steps for doing the same.

  Click on → Tools → Medium Manager → Add new Medium → Open path of CodeDownload.dll file and select it → Click on Ok

5.2.3 Configure device with Code Loader interface

- Code download plugin need to access set-top box via serial communication. So serial connection need to be configured in order to bind the device with the interface of the plugin. Please follow below steps for doing the same.

  Click on → Edit → Edit Configuration Wizard → Configuration Wizard will be opened.

- Now here we need to create a new device, which we will have to bind with the free interface of Code loader, by dragging and dropping that device on any of available interface of our plugin on right side panel. This will open a user interface of our Code Loaders configuration user interface.

  Select → KAcode Loader → Select TFTP method → This will set up a UI on right side. Here we need to enter appropriate configuration details, which will be used while running test case.

- Serial Configuration: Here Serial Connection parameters need to be selected appropriately such as, COM Port : COM Port with which the STB is connected using serial cable.

- Baud Rate: Enter baud rate compatible with STB. In case of ThinClient box, 38400 should be selected.

- Parity: Odd should be selected for ThinClient box. Other parameters are fine as default one.

- Verify all entered data and press ok button. If it was successful then a prompt will be shown along with interface number saying that it got configured and saved successfully.

5.2.4 Create test case for TFTP based CDL

- In order to create a new test case for the Code downloader follow the process below. Click on → Create New Test Case → select Utilities → Select Code loader → Drag and drop Icon

5.2.5 FTP build file upload and run test case

- Verify/update FTP server configurations where you want to upload the build files. Click on → Browse Button → Open Build File → Click on Ok

5.2.6 Run-time Variable Support

TFTP based CDL is having support for run-time variable support. You can provide $VarName in all input fields

- This will show prompt saying FTP upload was successful or not. Follow this process for Application Objects as well.

- Run the test case, make sure to select the device which you configured when asked.

Chapter 6

Implementation

6.1 DAC Based Code Download Process

Once you execute, Mozilla Firefox instance is created in background and following sequence of actions are performed:

- By pass security exception of SSL
- Open DAC URL
- Login to DAC
- Refresh box in case of Terminal Index addressing type
- Creating platform Code Object and Application Objects
- Adding all Objects to RADD
- Creating Decoder Conditional in case of Broadcast addressing type
- Creating Code Suit
- Creating Code Suit Components and select services for each component of code suit.
- Adding Code Suit Object to RADD
- Creating Assignment
- Targeting Assignment according to selected Addressing Type.
- Verify Version by serial log analysis

6.2 DAC User Interface

- There are several sections for the user inputs. There are some default values already provided according to the usability of the team. And some inputs will be automatically fetched from the build files to be uploaded on FTP server.
- FTP Config: User can override the FTP server configuration parameters here.
- DAC Object: User can upload platform object build files from here, by clicking on browse button, same functionality is supported for Application Objects as well. Dac object details will be fetched from ".dat" file automatically.
- Unicast, multicast and broadcast addressing modes are supported.
- FTP functionality eliminates use of WinSCP tool, which earlier was being used to upload all build files on server. Thus, it eliminated the manual effort.

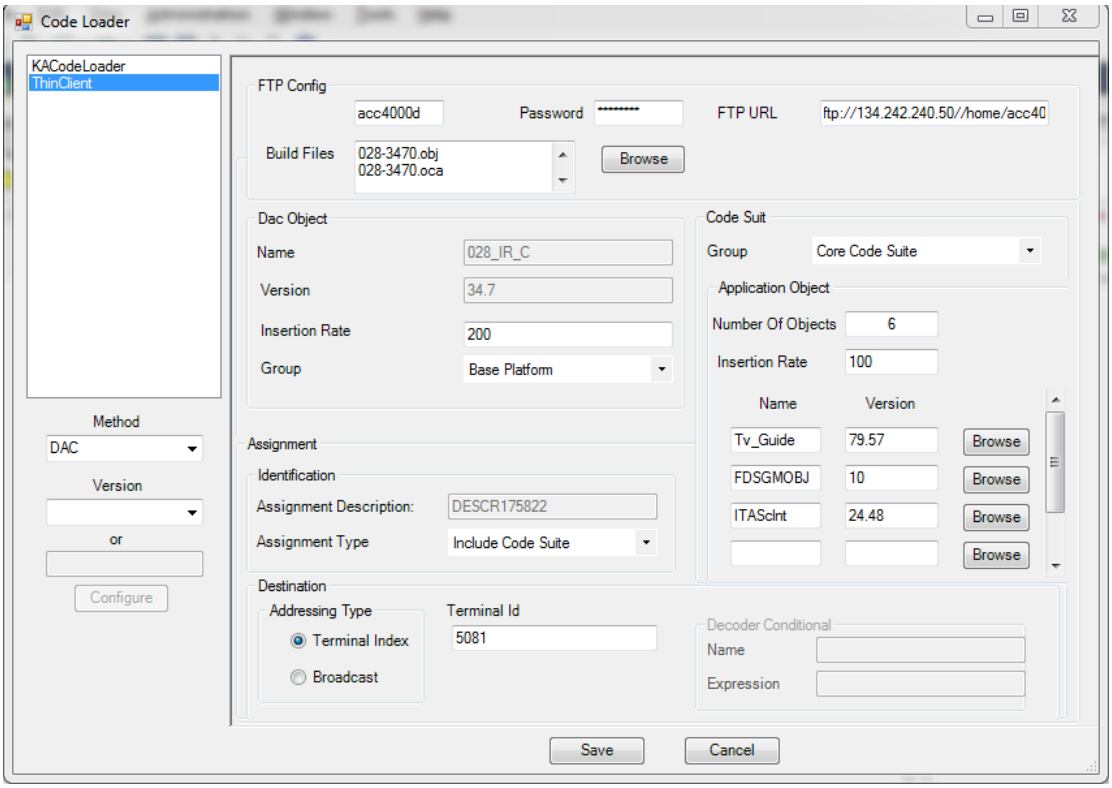


Figure 6.1: DAC UI With FTP and Multiple object support

6.3 Earlier server uploading was done Using WinSCP

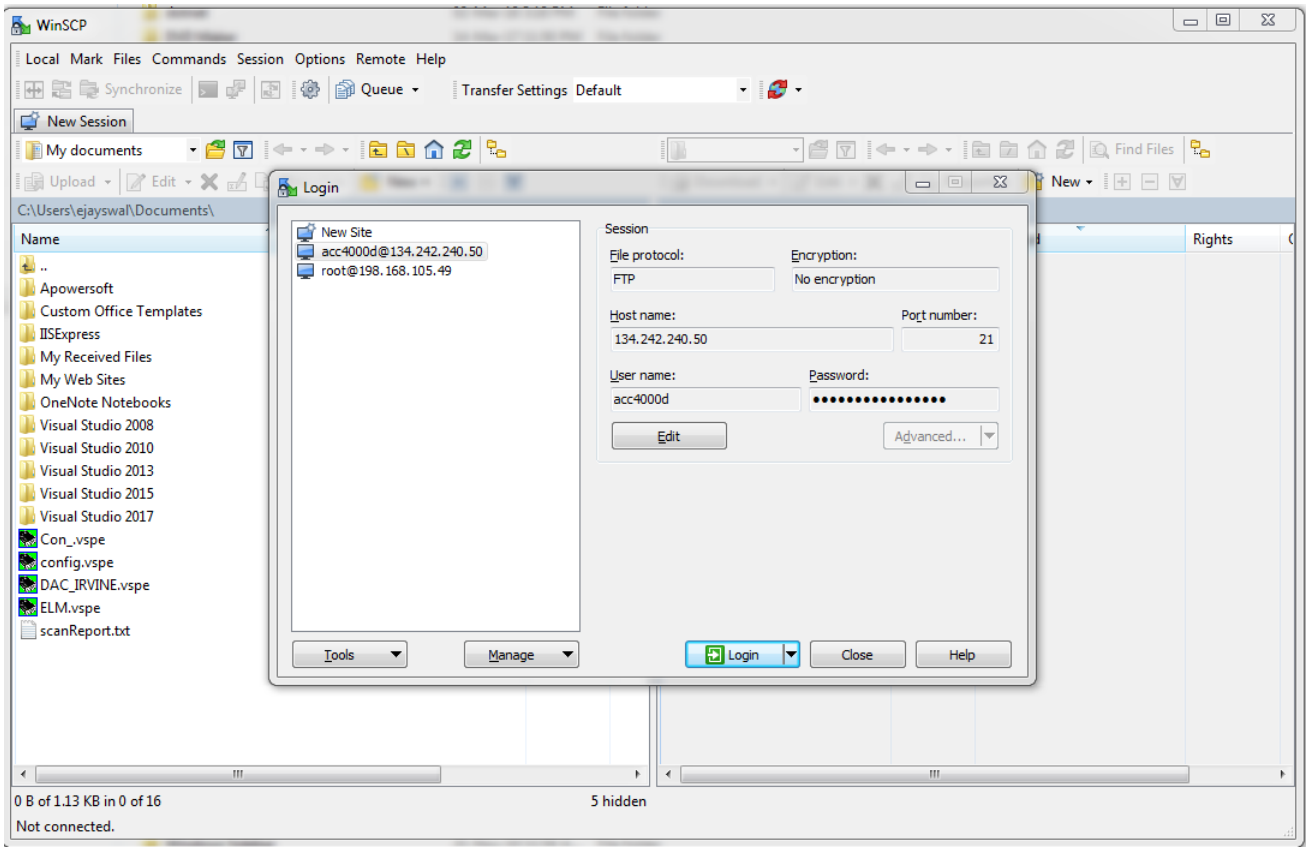


Figure 6.2: FTP using WinSCP

6.4 DAC Code Structure

Here is the Code structure which contains 3 basic classes being used for the core functionalities.

1. DACLoader: Responsible for using and the functionalities exposed by other two classes.

2. SerialCommunication: Responsible for establishing serial connection, fetching and processing serial logs from STB's.

3. SeleniumHandling: Responsible for creating firefox instance in back-end and automating whole web process part using selenium web driver.

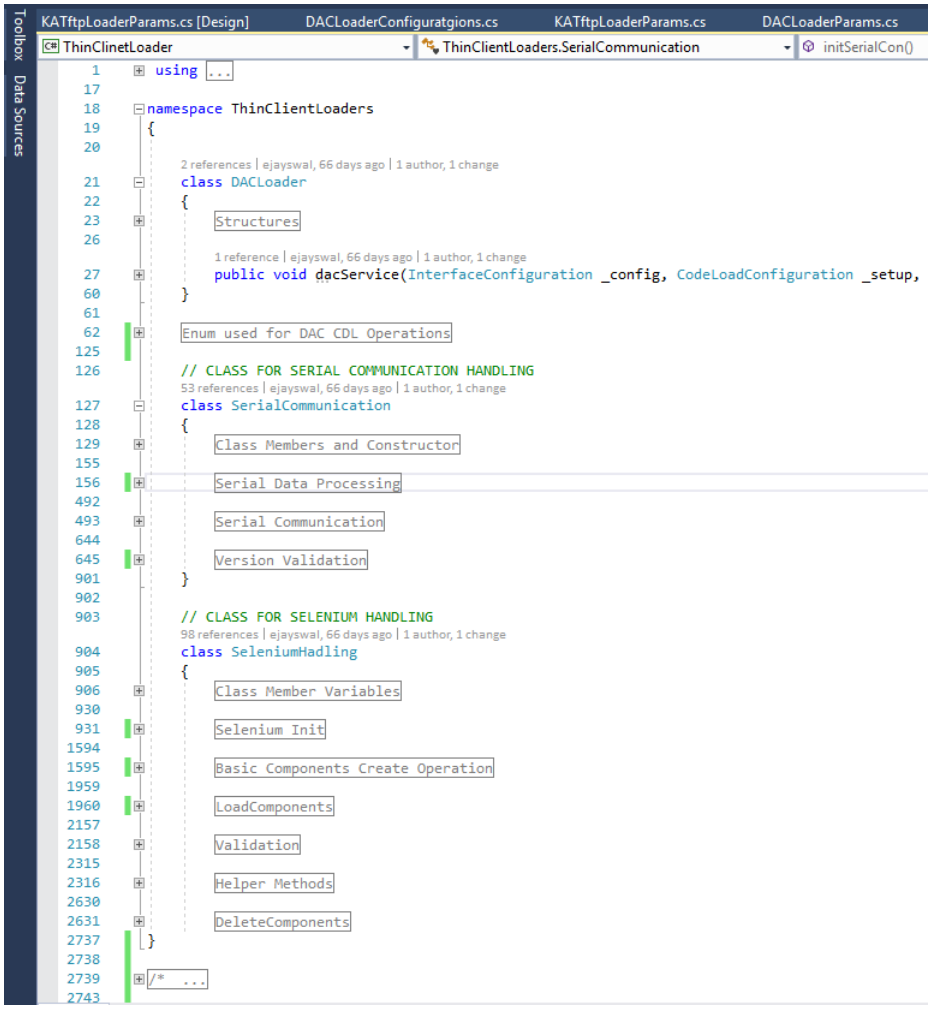


Figure 6.3: DAC Code Structure

6.5 Serial Communication code structure

This class contains methods and threads for establishing serial connection, fetching and processing serial logs from STB's.

Figure 6.4: Serial Communication Code Structure

6.6 Selenium Code Structure

This class contains methods for creating Firefox instance in back-end and automating whole web process part using selenium web driver.

It also have mechanism for version verification process by analyzing the serial logs fetched from the STB.

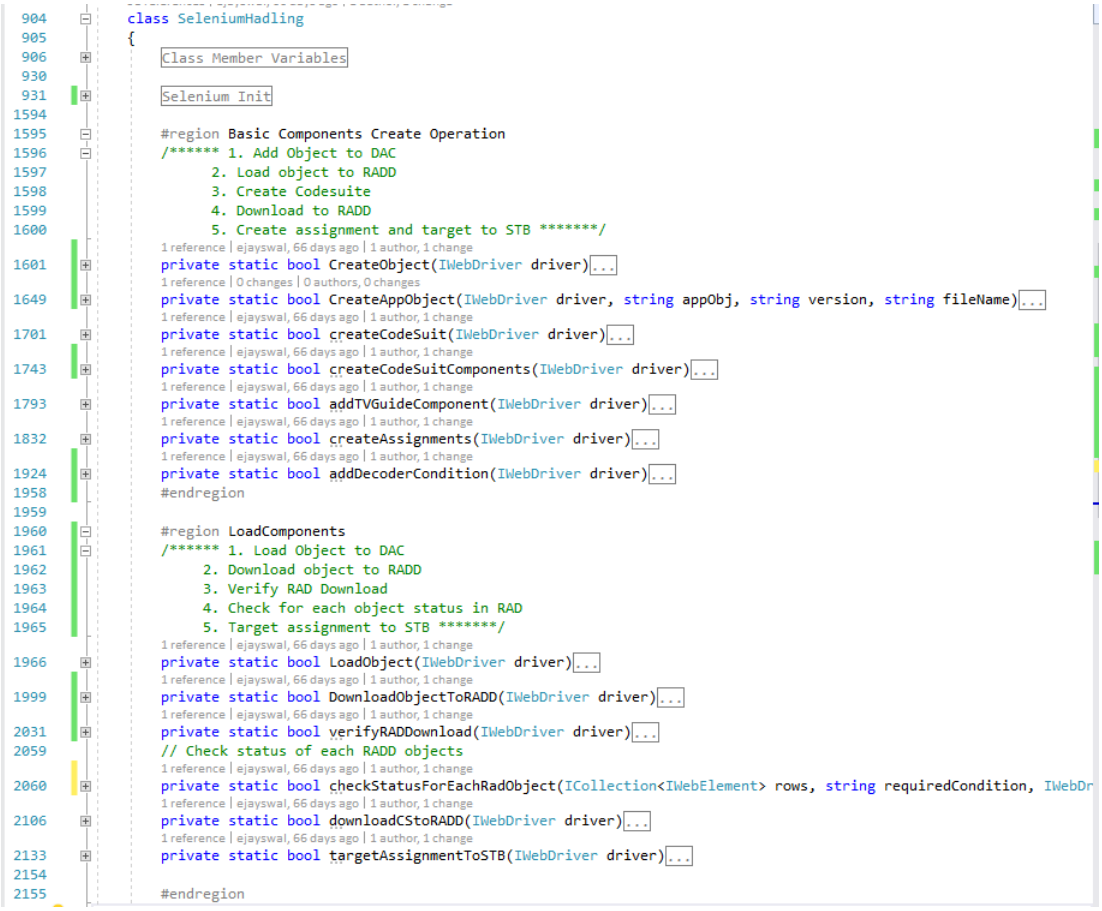


Figure 6.5: Selenium code structure

6.7 Creating Firefox Instance code snippet

Firefox instance need to be created which provide Security Certificate by pass, should be running in background etc. Here is the code snippet.

```
1499    private static IWebDriver CreateFirefoxInstance()
1500    {
1501        IWebDriver driver = null;
1502        FirefoxProfileManager ffProfileIni = new FirefoxProfileManager();
1503        FirefoxProfile profile;
1504        try
1505        {
1506            if ((profile = ffProfileIni.GetProfile("SeleniumFFProfile")) == null)
1507            {
1508                Console.WriteLine("SELENIUM PROFILE NOT FOUND");
1509                profile = new FirefoxProfile();
1510            }
1511        }
1512        catch(Exception ex)
1513        {
1514            throw new Exception("Firefox needs a Profile with \"SELENIUM\"");
1515            string reason = ex.ToString();
1516            ErrorHandling(reason, (int)EnumErrorCategory.SeleniumError);
1517        }
1518        profile.AcceptUntrustedCertificates = true;
1519        profile.AssumeUntrustedCertificateIssuer = true;
1520
1521        try
1522        {
1523            FirefoxOptions options = new FirefoxOptions();
1524            options.AddArguments("--headless");
1525            options.Profile = profile;
1526            driver = new FirefoxDriver(options);
1527        }
1528        catch (Exception ex)
1529        {
1530            string reason = ex.ToString();
1531            ErrorHandling(reason, (int)EnumErrorCategory.SeleniumError);
1532        }
1533        return driver;
1534    }
1535
```

Figure 6.6: Creating Firefox Instance code snippet

6.8 TFTP based CDL: Support FTP and Run-time variable

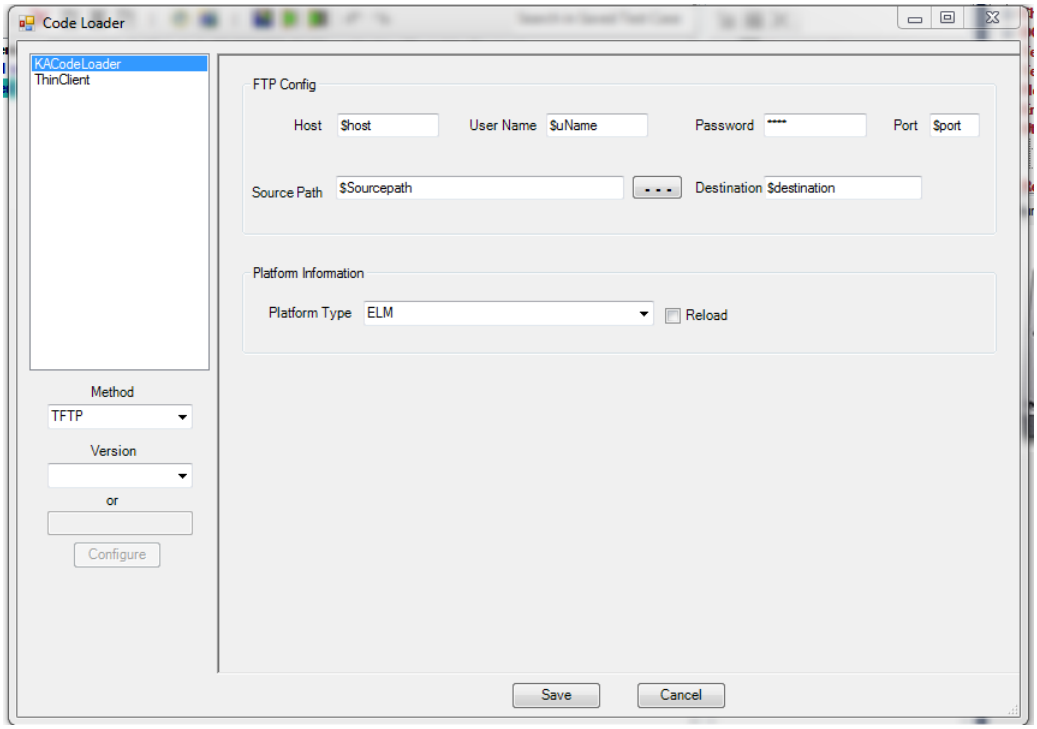


Figure 6.7: TFTP UI

6.9 TFTP UI: Runtime variable support

Runtime variable is the way to change values on run time without interacting eith the User Interface of plugin. Tester can even save the values of all run time variables in ".van" file and can retrieve back again too.

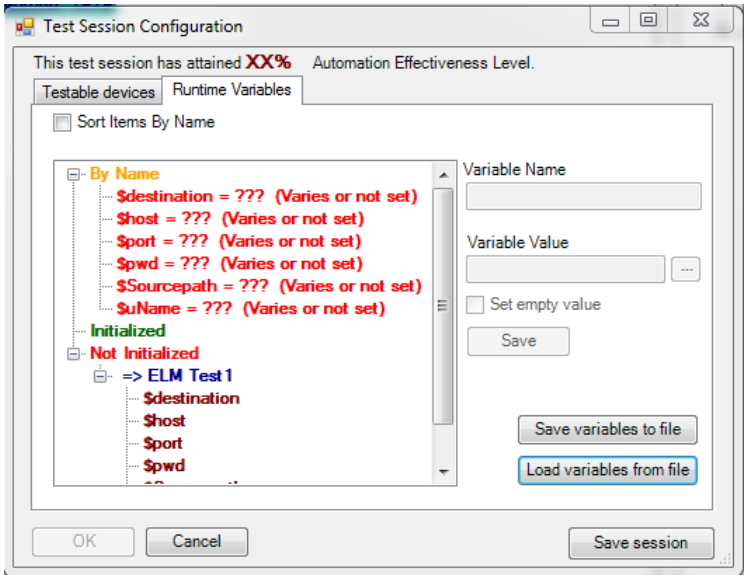


Figure 6.8: Run-time variable support in code downaload

6.10 TFTP Code Snippet

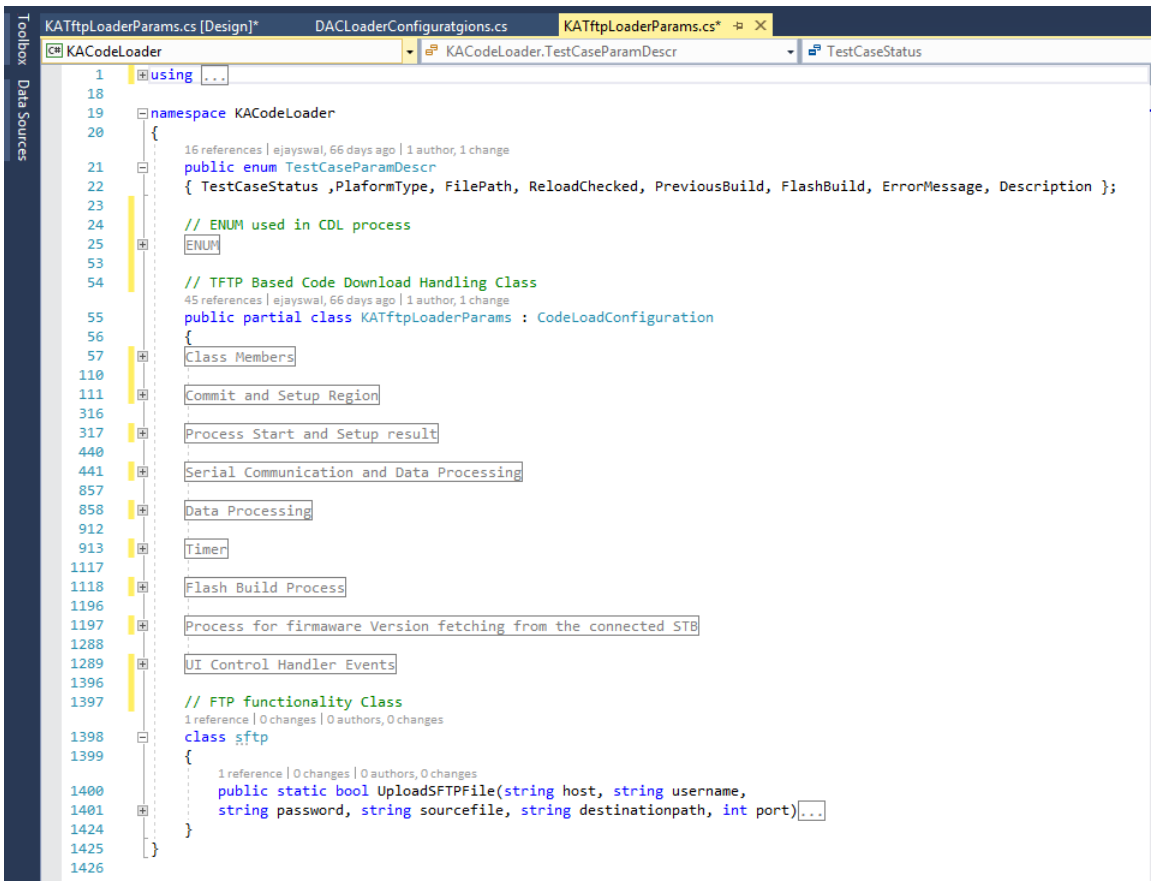


Figure 6.9: TFTP Code structure

Chapter 7

Other Work

7.1 ACCEL (Automatically Controlled Collection Of Error Logs) functionality enhancements.

7.1.1 Overview

- **ACCEL 2.0 System Overview**
  - Automatically Controlled Collection of Error Logs
  - ACCEL[7] is an Automated Solution that helps in the gathering of event related data from Settops or any device under test
  - The benefit of ACCEL lies in the fact that it helps automate the reporting of thousands of tests that would otherwise have to be reported manually
    - Cost savings by eliminating the manual reporting
    - Extreme time savings
    - Reporting error reduction
    - Standardized report format

7.1.2 ACCEL Components

- The following 5 subsystems make up ACCEL 2.0:

1. **AutoMOTO Medium**
   - This is the event data collection tool

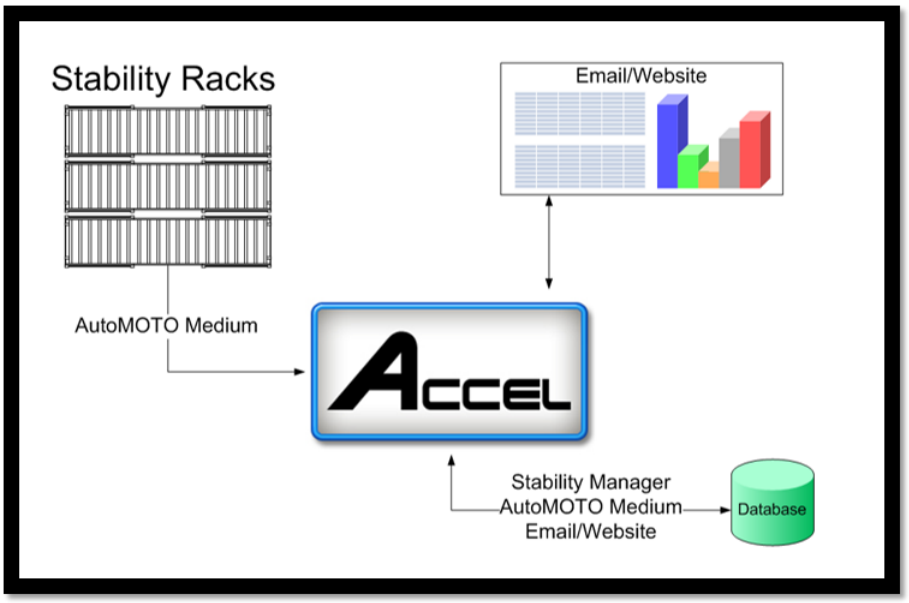


Figure 7.1: ACCEL System Overview

- Works with the device IDs under test that are contained in the Stability Manager

2. **Stability Manager**
   - A front-end GUI for the database
   - Information related to the devices under test need to be entered into the Stability Manager so automated reports can be generated
3. **Database**
   - Central Repository for all ACCEL 2.0 data
4. Report Email
   - Color coded for quick review
   - Sent out daily to a subscribed list by test group
   - Provides high level report statistics with links to drill-down to greater level of detail on the companion report website
5. Companion Report Website
   - Provides detailed information for the high level report communicated in the email

7.1.3 Tools and Technologies used

- **Microsoft Visual Studio(Asp.net)**:

  Microsoft Visual Studio is used as development environment of our plugin. Asp.net programming language is used for the development.[8]

- **ACCEL Website of ACCEL tool**

  It's a tool for automatically Controlled Collection Of Error Logs. It fetches serial logs form all the set-top boxes connected, does some analysis, generates report and sends it to the uses.[7]

- **AutoMoto**

  Version: 20.19. AutoMOTO is a framework, developed by ARRIS developers for their devices only. It can be used by developers as well as testers to automate test procedures, that otherwise would have consumed a lot of time.

  Using AutoMOTO, a tester can create, manage,schedule, and review the results of automated tests. The medium used is a .Net DLL program object which helps in communication between AutoMOTO software and various testable devices via a software or hardware interface (such as COM port, USB devices, TCP/IP and SNMP protocols etc.).[1]

- **MySQL Database**

  MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.[9]

7.1.4 Tasks and Description

1. Provide find link to get list of similar issues

   Accel report will have a result table generated based on the the input criteria selected. In case of failure, some cases will be repeated in the history so that they might already been logged in JIRA. So find link is provided in case if any match in the JIRA database is found, which will lead to redirecting to a web page displaying all such similar issues.

2. Filter out the retrieved list according to current Test Configuration Filtering need

to be applied to fetch only results which matches the test configuration of the currently selected issue. Filtering on fetching results is applied to get desired results and display the same on web pages.

3. Provide link to existing JIRA issues Identify if the issues already exist in database, then provide link to that particular JIRA issue. So that user can directly go on the particular issue and perform necessary actions.

7.1.5 Implementation and Screen-shots

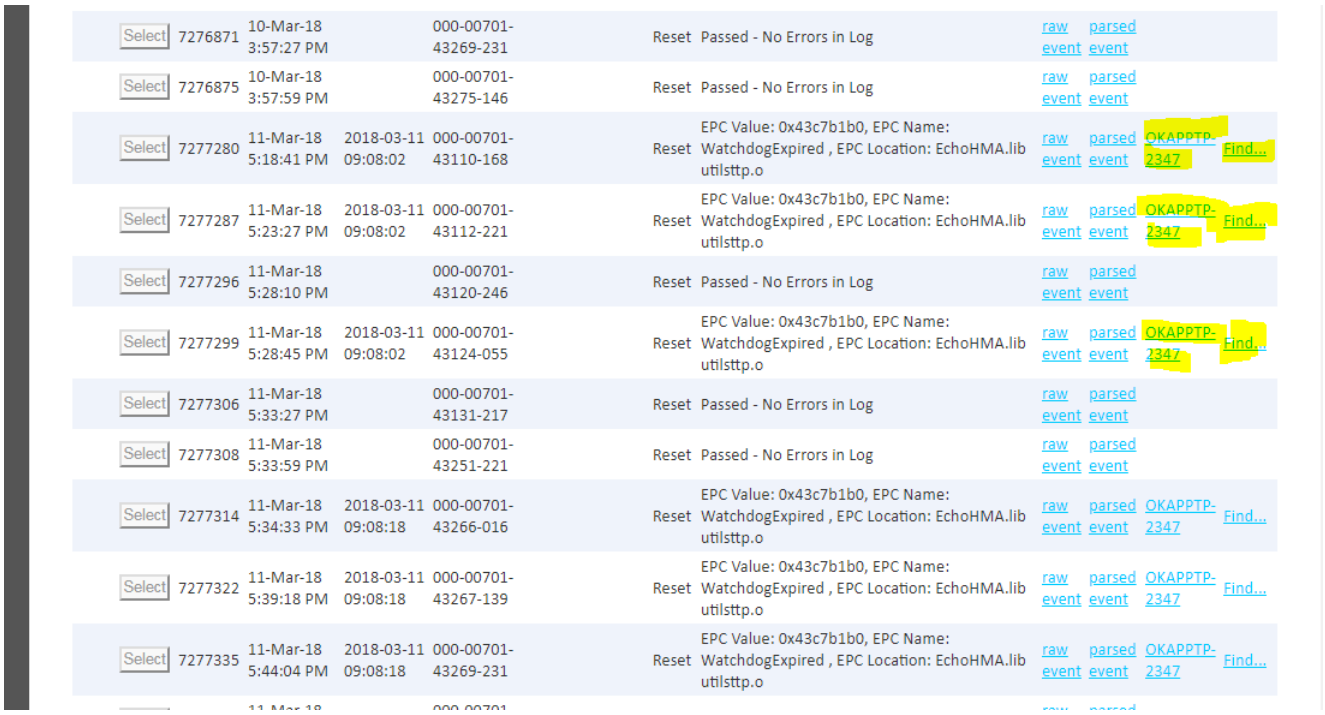


Figure 7.2: Find link to similar issues and JIRA link to known issues

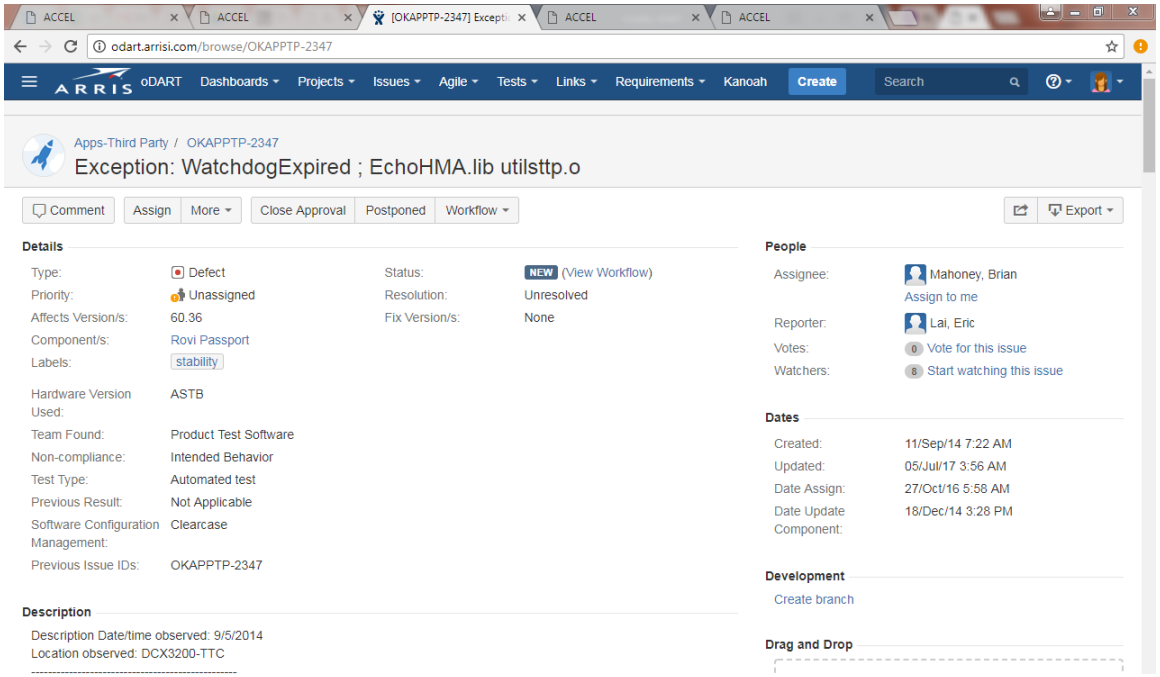


Figure 7.3: Output after clicking on find link

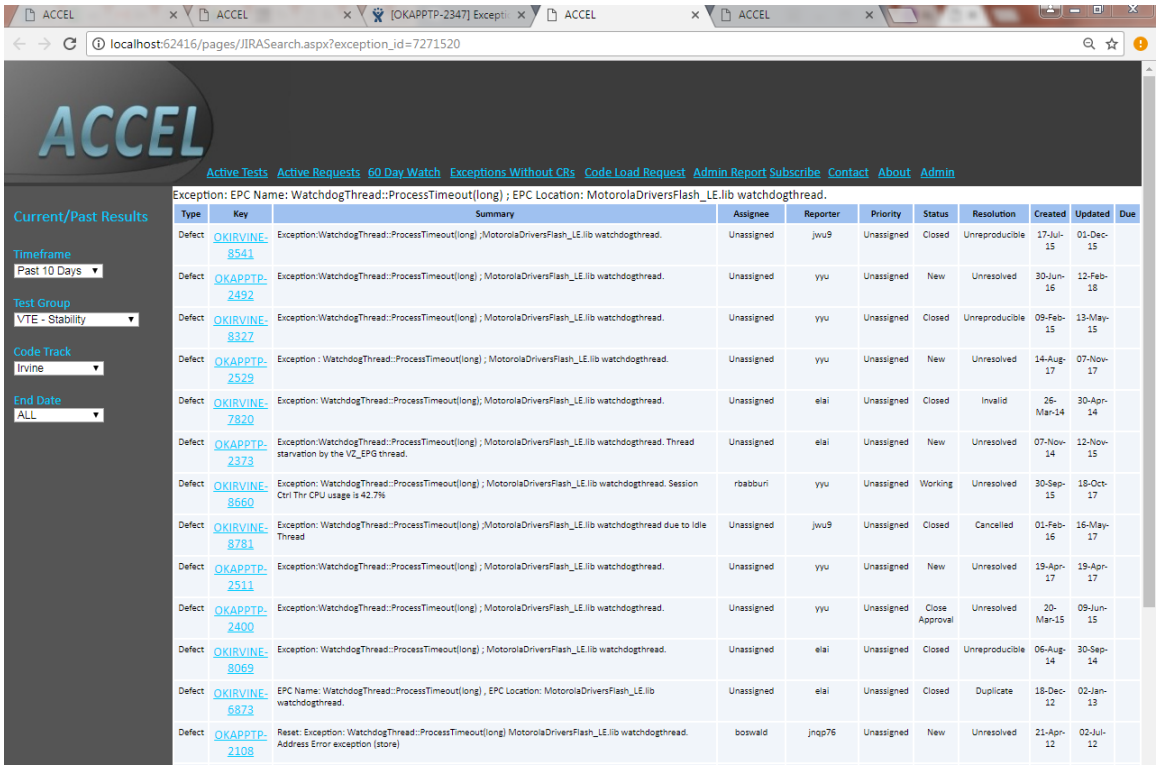


Figure 7.4: Known Issue's JIRA link

```
106             lblException.Text = epcNameLabel + r.EPCname + epcLocationLabel + r.EPClocation + "<br />";
107             string escapeEPCname = r.EPCname.Trim().Replace(@"\", @"\\").Replace("(", @"\(").Replace(")
108             string escapeEPClocation = r.EPClocation.Trim().Replace(@"\", @"\\").Replace("(", @"\(").Re
109             string query = "\"" + escapeEPCname + "\" AND \"" + escapeEPClocation + "\"";
110             DataSet jiraDataSet = new DataSet();
111             jiraDataSet.Tables.Add();
112
113             try
114             {
115                 RemoteIssue[] ris = j.getIssuesFromTextSearch(auth, query);
116                 String dispalyMessage = "";
117                 if (ris.Length.Equals(0))
118                 {
119                     var builder = new StringBuilder();
120                     builder.Append(lblException.Text);
121                     dispalyMessage = "No matching JIRA results found!";
122                     builder.Append('\n' + dispalyMessage);
123                     lblException.Text = builder.ToString();
124                 }
125                 string[] headers = {"Type","Key","Summary","Assignee","Reporter","Priority",
126                                     "Status","Resolution","Created","Updated","Due"};
127
128                 foreach (string header in headers)
129                 {
130                     jiraDataSet.Tables[0].Columns.Add(header);
131                 }
132
```

Figure 7.5: Code snippet to fetch data

7.2 APPIUM Mobile Automation

7.2.1 Overview

Make Mobile Automation generic for multiple operating systems and differently capable hard-wares. Basically the scenario was, company had all the setup of the automation already setup on MAC machine, now the task is to port whole automation set up on windows machine.

Appium tool is being used for the mobile application testing automation, along with the legacy software of the company called DATE. DATE framework is specially developed for automation testing, it supports .tcl scripts which are later convered into Ruby scripts. These ruby scripts are being used to run all the test cases on android as well as iOS mobile phones.This project required the basic understanding of APPIUM, Ruby scripts, mobile automation, android an iOS.

7.2.2 Tools and Technologies used

1 **Appium**

Appium is an open source test automation framework for use with native, hybrid and mobile web apps. [10] It drives iOS, Android, and Windows apps using the WebDriver protocol.

2 **Ruby**

Ruby is a reflective,dynamic,object-oriented, general-purpose programming language, which is being used for the test step automation.[11]

3 **iOS and Android mobile applications**

- **The ARRIS SURFboard Manager mobile application:** It steps you through configuring your SURFboard Wi-Fi home network using your iOS or Android mobile device (smartphone, tablet). With this mobile app, you can set up and monitor Wi-Fi access, Parental Controls, and security protection on all the connected devices (e.g., smartphones, tablets, computers, Smart TVs, gaming consoles, etc.) on your Wi-Fi home network.[12]

- **ARRIS HOMEASSURE:** It aims delivering reliable, high-bandwidth Wi-Fi to every corner of the home with a simple consumer experience.[13]

7.2.3 Tasks and Description

Tasks are are as follows:

1 Bring up Appium and all setup on Windows machine for Android.
Android and APPIUM bring up is done successfully. All the ruby tests are being executed using APPIUM on windows machine.

2 Explore to bring up Appium and all setup on Windows machine for iOS and check feasibility or alternative solution.
iOS and APPIUM has having so much of limitation of compatibility. So directly that way is not feasible at all. But found a solution, that using VMWare, Xcode can be installed in visrtual machine. That way iOS device can be tested using windows machine only.

3 Retrieving software and hardware capabilities from connected device dynamically and launch UI accordingly. Develop a generic approach independent of mobile operating system and hardware capabilities.
Different android and iOs devices will have different hardware and software capabilities. Thus they behave differently for the same test cases. Currently they are handled by providing their configurations in a hard coded ways. But this need to be done in a generic way that task is in progress.

Chapter 8

Return On Investment (ROI)

8.1 ROI Table

This automation achieves saving 217 staff days for the first year and 860 staff days from the next year on wards.

It achieves, saving 21700$per year and$86000 money of the company from the next year on wards.

Here we are considering 100$as staff day cost.

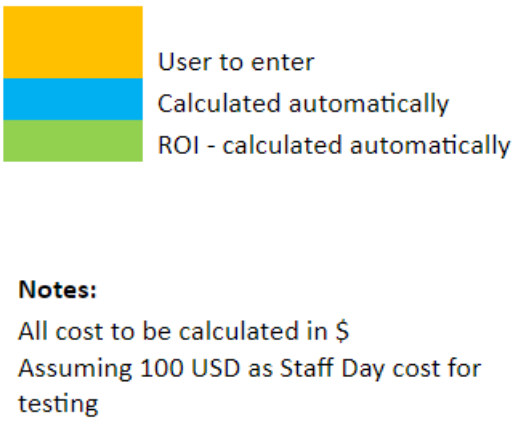


Figure 8.1: ROI representation

Sl no	Check list	Manual Testing	Automated Testing
1	No of staff days required to design test cases	0	45
2	Cost to design test cases	$0	$4,500
3	Cost of tool		$0
4	No of staff days required to automate the test cases		45
5	Cost to implement automation of test cases		$4,500
6	Total cost of automation		$9,000
7	No of staff days to execute the test cases	20	1
8	Cost to execute a full cycle of test cases (tester effort)	$2,000	$100
9	Number of cycles per year	15	15
10	Cost of testing per year	$30,000	$10,500
11	Saving for the first year		$19,500
12	No of staff days required fixing and Porting (in the subsequent year)		30
13	Maintenance Cost(Fixing and Porting effort)		$3,000
14	Cost of subsequent year	$30,000	$1,200
15	Saving for the subsequent year		$25,800
16	Number of cycles for subsequent year(including the cycles on the ported Platform)	15	12
17	ROI for the first year(benefit/Cost of Automation)		217
18	ROI for the subsequent year		860

Figure 8.2: Comparison of Manual Process and Automated Process

Chapter 9

Conclusion

Code Download automation plugin is basically developed for ARRIS internal use only with the agenda of upgrading/downgrading multiple Set-top boxes(STB) automatically through a common procedure. Different categories of STBs need to be targeted successfully based on the supported protocols for Code Download process such as DAC, TFTP etc and automated the whole process of firmware/application up-gradation and down-gradation.

Automating the procedure of downloading code on multiple set top boxes can save time and reduce man power required. Return on Investment suggest it saves a huge cost to the company and plays a vital role in increasing efficiency and throughput of the product development process.

ACCEL (Automatically Controlled Collection Of Error Logs) functionality enhancements are done successfully. Appium bring up with ruby support for android was done successfully along with successful test case execution. iOS setup for the same requirements is explored fully and report was submitted to company for their future reference.

Bibliography

[1] A. G. Inc., "Automoto documents." https://AutoMOTO.mot.com, 2018.

[2] M. Pikryl, "Winscp." https://winscp.net/eng, 2018.

[3] T. Teranishi, "Tera term." https://en.wikipedia.org/wiki/TeraTerm, 2018.

[4] J. Evans, "Selenium webdriver." https://www.seleniumhq.org/, 2018.

[5] M. C. Mozilla Foundation, "Mozzila firefox web browser." mozilla.org/en-US/firefox/new, 2018.

[6] A. G. Inc., "Arris internal documents." https://ArrisInternalDocuments365-095-32271x2.docx, 2018.

[7] A. G. Inc., "Accel documnets." https://ACCEL.mot.com, 2018.

[8] M. Corporation, "Microsoft visual studio." https://www.visualstudio.com/thank-you-downloading-visual-studio/?sku=Community&rel=15, 2018.

[9] O. Corporation, "Mysql database." https://www.mysql.com/downloads/, 2018.

[10] A. D. Group, "Tera term." https://appium.io/docs/en/about-appium/intro/, 2018.

[11] e. a. Yukihiro Matsumoto, "Ruby programming language." https://www.ruby-lang.org/en/, 2018.

[12] A. G. Inc., "Surfboard manager mobile application." https://ArrisInternalDocument365-095-32808_x.1_SURFboard_Manager_UG, 2018.

[13] A. G. Inc., "Arris home assure." http://www.arris.com/solutions/wi-fi-connected-home/, 2018.