

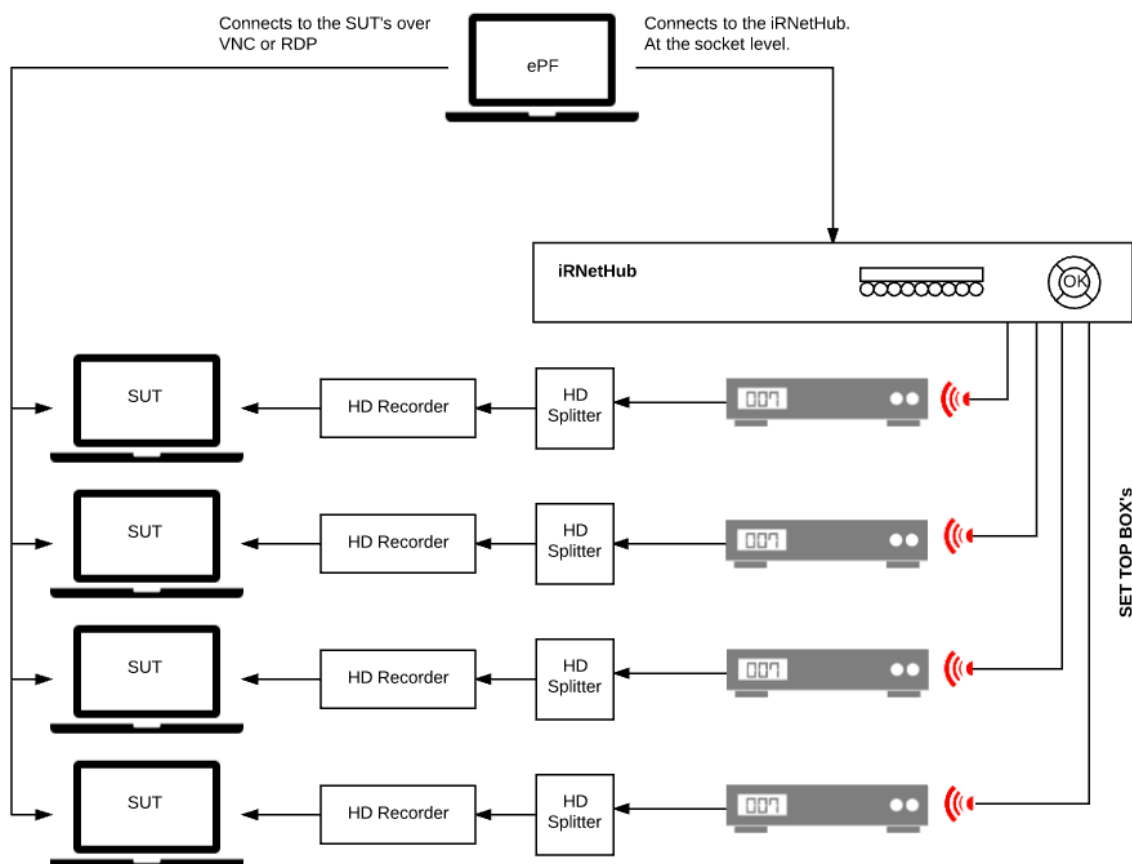


# Set Top Box Accelerator

V1.0

## Introduction

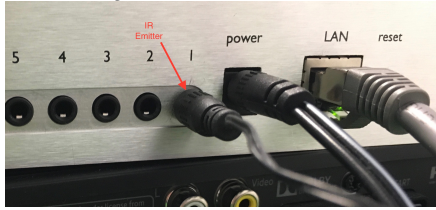

When it comes to automation testing against devices such as BT, YouView and Roku, it can be a difficult task as these devices are shipped with proprietary operating systems. However, many of the Set Top Boxes can be controlled via LIRC or WinLIRC, more commonly known as infrared signals. This article describes how to setup eggPlant Functional to automate application testing on the aforementioned products, as well as describe how using the technologies we have mentioned can be used to scale up the solution.



## Setup

This solution provides companies who are testing Set Top Boxes to scale up effectively. However, some parts of the solution can work just as effectively if replaced. This flexibility has been outlined in the table below:


Hardware		
Device Name	Description	Flexibility for alternative
iRNetHub	This hub enables you to transmit Infrared signals to the Set Top Box. The capacity of this device enables you to scale up to automate tests against 16 Set Top Boxes per hub.	Required... Link: <a href="http://www.redrat.co.uk/products/irnetbox-iv/">http://www.redrat.co.uk/products/irnetbox-iv/</a>

Infrared Transmitter	<p>These leads are required for the Infrared signal to be delivered to the Set Top Box's infrared receiver</p>	<p>There is flexibility here. However, there must be an IR Transmitter connected into the RJ45 jack on the iRNetHub.</p>  <p>Remember to position the IR emitter of the IR transmitter to point to the Set Top Box's IR receiver.</p> 
System Under Test (SUT)	<p>The SUT is used to confirm the state of the Set Top Box when instructions are sent from the iRNetHub.</p> <p>In the event you have a business requirement whereby you want to test Set Top Boxes in parallel, then multiple SUTs would be required.</p>	<p>Flexibility: Yes</p> <p>This will largely depend on the HDMI Recording device you decide on using with your the SUT. The SUT we have used is a Windows machine.</p>
HDMI Recorder	<p>This recording device will enable the visual output of the Set Top Box to be visible on the System Under Test.</p> <p>In the event you have a business requirement whereby you want to test Set Top Boxes in parallel, then multiple HDMI Recorders would be required.</p>	<p>Flexibility: Yes</p> <p>We have used one provided by AverMedia Extreme U 3.0(<a href="#">Click here</a>).</p>
HDMI Splitter	<p>This device decodes the signal output from the Set Top Box and sends it to the HDMI recorder.</p> <p>In the event you have a business requirement whereby you want to test Set Top Boxes in parallel, then multiple HDMI Splitters would be required.</p>	<p>Flexibility: Yes</p> <p>There are many HDMI Splitters on the market and should operate in the same way.</p>
HDMI Cables	<p>Required to connect the solution together. A single setup typically required 2 or 3. So if you are scaling up then the number of HDMI cables will increase.</p>	

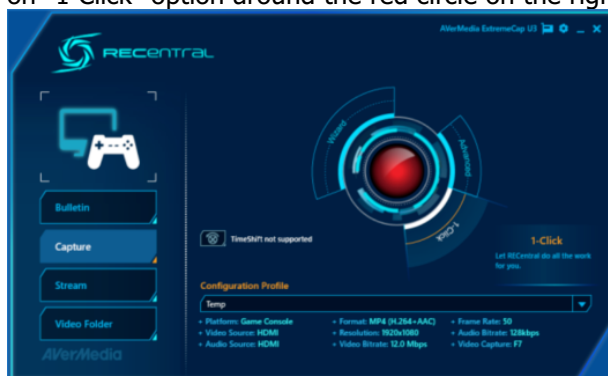
Network Cables	Required to connect the iRNetHub and the SUT to your test network	
<b>Additional Hardware – Optional</b>		
HDMI Hub	<p>In the event that you are run tests against multiple Set Top Box in a sequential manner, then having this device will enable to you switch between different Set Top Boxes.</p> <p>Note, most HDMI hubs typically come with a remote-control that allows you to switch the device you are viewing. To automate this you would need to use a Port on the iRNetHub to achieve this.</p>	

<b>Software</b>	
<b>Name</b>	<b>Description</b>
Python	We have created a couple of Python files that you will be required to use to connect to the iRNet Hub and send remote control signals to the Set Top Box.
HDMI Recorder Software	The HDMI Record should come with software that is installed on the System Under Test.
DBUtility	Provided by Redrat (the provider of the iRNetHub), this utility is only available on Windows. However, the XML file generated by this device can be moved and used away from the utility. This program enables you to create remote control profiles/records in the database. <a href="#">Click here</a> to download

## System Under Test Setup

**AverMedia RECentral:** This application is where the output of the Set Top Box will appear. To open the application look for the following icon:  (you can also find this application by clicking on the windows start icon and scrolling down the menu screen)

- Once open, click on "capture option" on the left-hand side of the application menu. Then click on "1 Click" option around the red circle on the right.



- On the 1-Click screen select games console on the left-hand side and press the "Detect" button. If you see a no signal message switch on the Set Top Box and press detect again



- Click on the ready button once available and maximize the screen if it doesn't default to full screen view.

## Windows Walk Through Guide

For this walk through please go through the following check list before running the eggPlant Functional scripts.

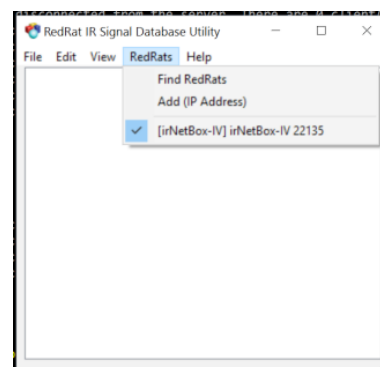
- ☐ The Hardware specified in the setup section had been acquired and installed
- ☐ Python is installed on the controller machine
- ☐ The RedRat Hub application is installed on the Controller Machine (<http://www.redrat.co.uk/software/redrat-hub/>)
- ☐ DBUtility provided by RedRat has been installed successfully (<http://www.redrat.co.uk/software/ir-signal-database-utility/>)

## Create your Dataset

Please follow the instructions below to and create a Dataset.

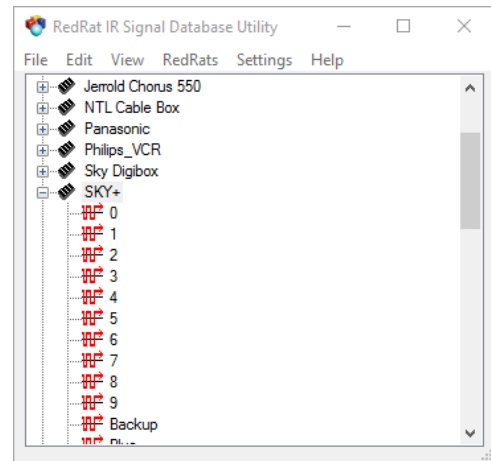
The process of creating a dataset from your remote-control unit is often referred to as *learning* or *capture*.

When opening the Database Signal Utility, the application will try to locate the iRNetHub. If it fails to do so then navigate to RedRats > Find RedRats.



To create a new dataset, follow the instructions below:

1. Create a new device/remote node from the *Edit* menu
2. Add a signal to the device, which will bring up a dialog box. Press *Learn IR*, and point your remote control at the IR input on the iRNetHub. For most types of IR signal, one has to press the remote-control button twice.
3. Test the quality of the captured IR signal by outputting it again to see if it will control your TV/STB. This is done by selecting the signal and using the *Edit > Test Signal Output* menu item



This process can be quite laborious, so the wizard can be used instead. This is started from the *Edit > New Device/Remote Wizard* menu item, and allows you to edit the list of IR signals to capture, then starts a loop for capture of them all

If you are testing one of the following Set Top Boxes, you can download the dataset that we have provided below:

- BT YouView Box

Once the Dataset is created copy the dataset from its current location and place it in the directory where the "RedRatHubCmd.exe" resides on your controller machine

In the same directory, create a Python folder and place the following file named TSignal.py in it.

## Walk Through Guide to using the Library

- Download the iRNetBox.suite library from the supporting library section below.
- Change the global variables in the "SendCommand" script
  - **iRHubIP:** This global variable refers to the IP address of your iRNetHub's IP address.
  - **ControllerIP:** This global variable refers to the IP address of the controller. If this is not set correctly then you will not be able to send commands to the iRNetHub.
  - **ChannelID:** This global variable refers to the channel where the infrared transmitter lead is plugged into the iRNetHub.
- Create a new eggPlant Functional suite and add the modified iRNetBox.suite as a helper suite to your newly created suite. More on helper suites [Click here](#)

Supporting Library & Files	
iRNetBox.suite	Within this suite there are a number of modules & functions that will enable you to create scripts that will automate your test cases.
RedRatHub.py	Used by eggPlant Functional as part of the process to Open a Socket session with the iRNetHub
iRNetBox_Run.bat	
KillSwitch.bat	Used to terminate the socket session between the controller machine and the iRNetHub
TSignal.py	Used by eggPlant Functional as part of the process to send the remote-control signal to the Set Top Box
SendSignal.bat	
Library Functions/Modules	
OpenSession	
params	<a href="#">ConfigFile</a>

```
//the following code opens the batch file and opens the connection into the iRNetBox
set the directory to the folder of the folder of me

//used the shell command (internal functionality of eggPlant functional to trigger the batch file that
opens the session)
set the shellcommand to "ShellExecute"
put shell("iRNetBox_Run.bat", ConfigFile)
wait 10
```

***To use this script.***

In your suite call the function at the beginning of your script so that the session is open to receive commands. To do this, type the following:

OpenSession "STBs.xml" // you must specify the dataset that you created in an earlier step between the quotes

### CloseSession

```
//the following code opens the batch file and opens the connection into the iRNetBox
set the directory to the folder of the folder of me

//uses the shell command (internal functionality of eggPlant functional to trigger the batch file that
Terminates the session)
set the shellcommand to "ShellExecute"
put shell ("KillSwitch.bat")
log "Session ended..."
wait 3
```

***To use this script.***

In your suite call the function at the end of your script so that the session is terminated the OpenSession function. To do this, type the following:

CloseSession

### SendCommand

```
//takes the value from your script and maps this across to the variable ButtonPress
params ButtonPress

put "10.145.128.132" into global iRHubIP
put "10.145.128.50" into global ControllerIP
put "01" into global ChannelID // should you have more devices then this needs to be set at the
user instruction level

//the following code opens the batch file and opens the connection into the iRNetBox
set the directory to the folder of the folder of me

//uses the shell command to send the variables to the Set Top Box
set the shellcommand to "ShellExecute"
put shell("SendSignal.bat", global ControllerIP && global iRHubIP && ButtonPress && global
ChannelID)
wait 0.02
```

***To use this script.***

In your suite call the function when you want to send an instruction to the Set Top Box. Essentially this script acts as your remote control

SendCommand "YouView" // you must specify the name you gave the signal in the dataset. "YouView" is the name for the button on the remote control.

#### screenPart

**params** **rec** -- Values are between 0 and 1 and represent the proportion you want to go in the x and y direction for the top left and bottom right coordinates.

**put** (MsgBox:"0.28,0,0.72,1",Movie:"0,0.2,1,0.7",SearchArea:"0,0.5,1,0.8") **into** **RectangleList**

**return** (item 1 of **RectangleList**.(**Rec**) \* **ConnectionInfo**().**ScreenSize.x**,item 2 of **RectangleList**.(**Rec**) \* **ConnectionInfo**().**ScreenSize.y**, Item 3 of **RectangleList**.(**Rec**) \* **ConnectionInfo**().**ScreenSize.x**, Item 4 of **RectangleList**.(**Rec**) \* **ConnectionInfo**().**ScreenSize.y**)

#### *To use this script.*

In your suite, call the function when using the searchrectangle command.

Set the searchrectangle to screenPart("SearchArea") // SearchArea refers to the mapped information within RectangleList in the screenPart function

#### DeviceState

**params** **Cmd**

SendCommand "Power"

if **Cmd** = "On" then

    waitfor 50, text:"No TV Channels currently available"

    log "Device is ready to start tests"

else

    waitfor 20, "NoSignal"

    log "Device has closed down gracefully"

end if

#### *To use this script.*

This function allows you to use the power button on the remote-control to switch on or switch off the device.

DeviceState "On" // this function has two states "On" or "Off".

#### SelectButton

(see video User Journey – Read Details)

**params** **Button**

**repeat** 6 times

**repeat** 5 times

        SendCommand "Left"

**end repeat**

        SendCommand "Up"

**end repeat**

waitfor 20,"SelectedImage/TVChannels"

**put**(TVChannels:"0,0", PictureAndSound:"1,0", BroadBandConnection:"0,1", "TVSignalQuality":"0,2", Recordings: "1,1", PowerAndStandby: "1,2", MobileDevices:"1,3", RestrictRatedProgrammes:"2,0", ChangeParentalControlPIN:"2,1", LanguageAndSubtitles:"3,0", Appearance:"3,1", AudioDescription:"3,2", AudioFeedback:"3,3", DeviceInformation:"4,0", SoftwareInformation:"4,1", TermsOfUse:"4,2", FactoryReset:"4,3") **into** **ButtonMapping**

**put** **ButtonMapping**.(**Button**) **into** **UpandDown**

**repeat** item 1 of **UpandDown** times

    SendCommand "Down"

**end repeat**



```
repeat item 2 of UpandDown times
    SendCommand "Right"
end repeat
```

```
waitfor 20,"SelectedImage/"&Button
```

**To use this script.**

In your suite use the following syntax to find images of a menu on or off the screen. You must first save an image and add the image name to "ButtonMapping" list. Then All you need to do is pass the image reference name from your script to the SelectButton Function.

```
SelectButton "PictureAndSound"
```

<b>PlayValidationVerification</b>	(see video User Journey – Play Video from App)
-----------------------------------	--

```
params timeinSec,VideoImgName
put SuiteInfo().imagesfolder&"/" into MyImageLocation
```

```
set PlayFailed to 0
```

```
CaptureScreen (Name: MyImageLocation&VideoImgName, Rectangle: screenpart("Movie"))
```

```
Repeat for timeinSec seconds
```

```
    if not imagefound(VideoImgName) then
        CaptureScreen (Name: MyImageLocation&VideoImgName, Rectangle:
screenpart("Movie"))
    else
```

```
        if imagefound(VideoImgName) then
            set PlayFailed to PlayFailed+1
            CaptureScreen
```

```
        else
            set PlayFailed to 0
        end if
    end if
```

```
End repeat
```

```
If PlayFailed > 0 then
    LogWarning "The video stopped playing..."
```

```
else
    LogSuccess "Video Playing...."
End If
```

**To use this script.**

In your suite use the following syntax to verify that a video is playing successfully or not. When calling this function, you must pass in 2 parameters. The first parameter specifies the length of time in seconds you want to run the test; the second parameter refers to the name you need to give the live image capture used by PlayVideoValidation to verify if the video is playing.

```
playVideoValidation "20","VideoImageCaptured"
```

**Additional documentation**

Project Plan	Here is a template file of a project Plan that gives you an indication on how long it would typically take to install the hardware and setup your test automation solution for Set Top Boxes
--------------	--

Examples of how user journeys can be written	<ul style="list-style-type: none"><li>- User Journey – Play Video from App</li><li>- User Journey – Read Details</li></ul>
--	--