**eTRAC APP Package Description - Android**

eTRAC Android application has following packages in its project,

1. com.viasat.etrac

2. com.viasat.etrac.adapters

3. com.viasat.etrac.common

4. com.viasat.etrac.controls

5. com.viasat.etrac.dataobj

6. com.viasat.etrac.listeners

7. com.viasat.etrac.services

8. com.viasat.etrac.utils

**1. com.viasat.etrac:**

This is the main package of eTRAC Application. Here in this package following classes are there,

ViaSatEtracStart

InstructionsActivity

IpCheckActivity

IpErrorActivity

RegistrationActivity

TestMenuActivity

SpeedTestActivity

WebPageLoadTimerActivity

ThroughputActivity

VideoLoadTestActivity

TestResultActivity

TestDataCompareActivity

**ViaSatEtracStart:**

This is the App launcher activity class, from this class only application will starts when user launches the application into the device. Here in this class it will shows one screen which display with ViaSat Logo and application name as eTRAC Application. This screen will displays for 3 seconds, after 3 seconds it goes to InstructionsActivity class if the app is first time launching or user not accepted ViaSat Terms and Condition, otherwise it goes to IpCheckActivity class.

**InstructionsActivity:**

This class will shows one screen which will display ViaSat Terms and Condition in a webview along with two buttons Accept and Decline. If user accept the term and condition, then it goes to IpCheckActivity class. If user declines the ViaSat Terms and Condition, app display alert message and it stays on same screen until user accept the terms and condition. If user accepts ViaSat application terms and condition, then this class will not be called in subsequent app launch.

**IpCheckActivity:**

This class start check internet availability on the device by calling isNetworkAvailable(Context context) method in NetworkUtils class. While it checks internet in background, it display message to user as “Checking for internet connection...”. If the device is connected to valid network then it goes calls RegistrationActivity class, otherwise it calls IpErrorActivity class along with the network failure type.

**IpErrorActivity:**

This class will shows alert message to user depending on the type of network failure, passed by IpCheckActivity class. If devices is not connected to any wifi network,this screen appear to user with an alert message shows that “Device is not connected to Wifi Router. Do you want to enable WIFI?”, along with two options “OK” And “CANCEL”. If user clicks “CANCEL” then alert message will be dismisses, and stays in the same screen. If user clicks “OK”, alert message will be dismisses and navigates the user to devices default WIFI Settings Screen, If devices is connected to wifi network and if no response has received with in 30sec, its display alert message to user as “Internet connection is not available”, along with “OK” option, if user clicks “OK” alert message will be dismisses, and stays in the same screen. Here in this screen there is another button ie. “Try Again”, by clicking this button user can initiate the internet check again in this screen by calling isNetworkAvailable(Context context) method in NetworkUtils class. If internet check is success, it calls RegistrationActivity class.

**RegistrationActivity:**

This class will checks weather the gps is enabled are not by calling canGetLocation() method of LocationUtils class, if gps is not enable then it will show Alert message as “GPS not enable. Do You want to go to settings menu?”, along with two options as “Settings” which navigates the user to GPS Settings screen and the other is “Cancel” which dismisses the alert and stays in the same screen. If GPS enable then gets the current latitude and longitude by calling getLatitude() and getLongitude() methods of LocationUtils class and stores the values into SharedPreferences.

To perform various internet related test operation using eTRAC application, user has to register the flight by filling following details,

Email ID

Flight ID

Provider Name

Service Details

User Comments

To get these details from user this class shows a screen. This screen has two buttons, “Submit” & “Clear All”. Here in this screen Email ID, Flight ID and User Comments fields are used take user inputs from keypad, where as Provider Name and Service Details are dropdowns, to allow the user to select an option from available options.

User Comments field is optional, and Email ID field becomes as optional if selected Provider Name is “Anonymous”.

If user click on “Clear All”, all entered field will be cleared.

If user clicks on Submit button, app validates all fields.

If validation failed, app will show error message to user for updating correct field.

If validation got succeed, then internet check will runs in background by calling isNetworkAvailable(Context context) method in NetworkUtils class , if internet check fails then it shows appropriate alert to user, if internet check is also succeed then app will store all fields data into its device data base using SharedPreferences and also update fields data to ViaSat server by calling Server API which are exposed from ViaSat server, and calls TestMenuActivity class.

**TestMenuActivity:**

Through this class user can start following test cases,

Run All Tests

Speed Test

WebPage Load Timer

Send Byte Count Measurement

Throughput

Video Load Timer

Test Results

Router Details

Here in this class, all the test cases can be shown using ListView and TestMenuAdapter class, each List Item represents one test case. when user clicks on any List Item, it will checks the internet in background (except Test Results case) by calling isNetworkAvailable(Context context) method in NetworkUtils class., if internet check faills, then it shows appropriate alert message to user. If the Internet check is succeed then only corresponding test will runs.

If user is Navigated from RegistrationActivity class to TestMenuActivity, after 1Sec it will runs “Run All Tests” Case Automatically.

**Run All Tests:**

In this test case following test will be run sequentially and after completion of all tests, it will display the test result on Test Result screen.

Automated test cases are - Speed Test, Webpage Load Timer, Throughput Test (iPerf), and Video Test. While individual test are running under automated test cases, corresponding test class will be called.

While Test is running, if internet connection is not available, then automated test will be aborted and it will come back to TestMenuActivity and display alert message to user. User can abort this test at any stage by clicking Back arrow in individual test case, to come back to TestMenuActivity.

**Speed Test:**

To perform speed test app calls SpeedTestActivity class.

**WebPage Load Timer:**

To perform WebPage Load Timer test app calls WebPageLoadTimerActivity class.

**Send Byte Count Measurement:**

To get Send Byte Count Measurement, app invokes Android system call to get the device received byte count and transmitted byte count. To get these values app calls *getTotalRxBytes*() and *getTotalTxBytes*() methods of TrafficStats class and shows the result using CustomDialog.

**Throughput:**

To perform Throughput test app calls ThroughputActivity class.

**Video Load Timer:**

To perform Video Load Timer test app calls VideoLoadTestActivity class.

**Test Results:**

To check Test Results**,** app calls TestResultActivity class.

**Router Details:**

To get Router Details, app calls routerInfo(Context context) method of NetworkUtils and shows the router details using CustomDialog.

**SpeedTestActivity:**

To perform speed test, test invoke below HTML link from test logic and show the result on WebView.

Link: http://speedtest.via-satellite.net/etrac/speedof.html

This class will call startWifiCheck() method at the time of test starting, to check continuously internet availability using Threads. While test is performing, if internet check fails, test will be aborted and user will be redirected to TestMenuActivity and appropriate failure alert message will be shown to user, and there will be no update on test result.

In this class, link is loaded using a webview, and WebViewClient and WebChromeClient are set to the webview. Through WebViewClient app can get the completion of page loading. Afetr complet loading, to get the result from HTML link, below JavaScript is invoked from WebViewClient.

wvSpeedTest.loadUrl("<javascript:alert(document.getElementById('download').innerHTML>)");

wvSpeedTest.loadUrl("<javascript:alert(document.getElementById('upload').innerHTML>)");

wvSpeedTest.loadUrl("<javascript:alert(document.getElementById('ping').innerHTML>)");

wvSpeedTest.loadUrl("<javascript:alert(document.getElementById('jitter').innerHTML>)");

After calling these JavaScript functions, values can be get from WebChromeClient's onJsAlert(WebView view, String url, String message, JsResult result) method. After getting following details, the result will be shown using CustomDialog with “OK” button, and also update the result to device internal database and test result also published to ViaSat server using ServerAccess class. If user clicks on “OK”, SpeedTestActivity will be finishes and navigates the user to TestMenuActivity. If SpeedTestActivity is called under the RunAllTests case, After successful test completion, CustomDialog with result will shown for 1.5 sec, After 1.5 sec CustomDialog will be dismisses, SpeedTestActivity will be finishes and calls WebPageLoadTimerActivity class.

* Download Time
* Upload Time
* Ping Time
* Jitter

**WebPageLoadTimerActivity:**

To perform WebPageLoadTimer test, this class uses WebView to load each url, WebViewClient set to the WebView, and a ListView to all the urls.

This class will call startWifiCheck() method at the time of test starting, to check continuously internet availability using Threads. While test is performing, if internet check fails, test will be aborted and user will be redirected to TestMenuActivity and appropriate failure alert message will be shown to user, and there will be no update on test result.

After the startWifiCheck() method called, this class will calls GetWebLoadData AsyncTask to get the urls, which are need to be loaded in Webview from below link

Link: <http://speedtest.via-satellite.net/etrac/content/webPageTimer.json>

Response Message From ViaSat Server:

{

"response":"success",

"webPageTimer": {

"timeout":"120.0",

"numURLs":"10",

"url0":"http://www.wikipedia.org",

"url1":"http://www.about.com",

"url2":"http://www.google.com",

"url3":"http://www.linkedin.com",

"url4":"http://www.yahoo.com",

"url5":"http://www.yelp.com",

"url6":"http://www.pinterest.com",

"url7":"http://www.bing.com",

"url8":"http://www.viasat.com",

"url9":"http://www.outlook.com"

}

}

GetWebLoadData class will get the data from link and parses the responce data and returns the urls to WebPageLoadTimerActivity. Then WebPageLoadTimerActivity will bind the urls using a ListView in the UI. After binding urls with ListView, loading of each url in webview is starts one by one. Through WebViewClient app can identify the sart time and finish time of a url to calculate that url's load time, and app can also identify if any error has occured while loading a url. By using these details each urls status in ListView will be updated and loading of next url will be started, it will be a continues process until all urls are loaded. After all urls are loaded, this activity will calculates the loaded urls count, toatal urls load time and average urls load time, and these details will be shown as result using CustomDialog with “OK” button, and also update the result to device internal database and test result also published to ViaSat server using ServerAccess class. If user clicks on “OK”, WebPageLoadTimerActivity will be finishes and navigates the user to TestMenuActivity. If WebPageLoadTimerActivity is called under the RunAllTests case, After successful test completion, CustomDialog with result will shown for 1.5 sec, After 1.5 sec CustomDialog will be dismisses, WebPageLoadTimerActivity will be finishes and calls ThroughputActivity class.

Following data will be shown as result on successful test completion,

* Web Pages Loaded
* Total Load Time
* Avearge Load Time

**ThroughputActivity:**

This class will call startWifiCheck() method at the time of test starting, to check continuously internet availability using Threads. While test is performing, if internet check fails, test will be aborted and user will be redirected to TestMenuActivity and appropriate failure alert message will be shown to user, and there will be no update on test result.

To perform Throughput test, this class calls following link to get servers detail.

Link: <http://speedtest.via-satellite.net/etrac/content/iperf.json>

Response:

{

"response": "success",

"iperf3Servers": {

"numServers": "2",

"servers": [

{

"description": "iMAC(internal)",

"ip": "172.24.4.178",

"port": "5201",

"timeout":"30"

},

{

"description": "Amazon(cloud)",

"ip": "54.208.183.22",

"port": "5201",

"timeout":"60"

}

]

}

}

To get these details this class calls GetServiceData AsyncTask, which takes care of calling the above mentioned link and parses the response and binds the response data to a Popup Window through which user can select the server among the available servers to perform Throughput test. Here in this class, there is button “Run Test”, by clicking this button test can be started after selecting a server.

When user clicks on “Run Test” button, this class calls initIperf(ServerInfoDo infoDo) method, in which iperf file from assets is opens and writes into /data/data/com.viasat.etrac/iperf path if the file is not there, after that it calls ThroughputTest AsyncTask, which class is responsible for performing test and finally the result will be publishes to ThroughputActivity using StatusListener listener, basing on this listener ThroughputActivity will updates the test status to user using testStarted(), testStatusUpdated(String data),testCompleted() and testCanceled() methods. If the test compltes successfully, StatusListener will pass the result using testStatusUpdated method, then that result will be parsed and will be shown as result using CustomDialog with “OK” button, and also update the result to device internal database and test result also published to ViaSat server using Server Access class. If user clicks on “OK”, ThroughputActivity will be finishes and navigates the user to TestMenuActivity. If ThroughputActivity is called under the RunAllTests case, After successful test completion, CustomDialog with result will shown for 1.5 sec, After 1.5 sec CustomDialog will be dismisses, ThroughputActivity will be finishes and calls VideoLoadTestActivity class.

Following data will be shown as result on successful test completion,

* Interval
* Transfer
* Bandwidth

**VideoLoadTestActivity:**

To perform Video Load Timer test, this class uses WebView to load below link,

Link: <http://speedtest.via-satellite.net/etrac/youtube.html>.

This class will call startWifiCheck() method at the time of test starting, to check continuously internet availability using Threads.While test is performing, if internet check fails, test will be aborted and user will be redirected to TestMenuActivity and appropriate failure alert message will be shown to user, and there will be no update on test result.

Here in this class from onResume() method activity will calls startYoutube() method, which will take care of getting HTML data from the above mentioned link, loads that data into the WebView to load the video from a Thread. To get the completion of video, a WebViewClient is set to WebView, through this listener it can identify the video completion event, at that time it will calls the following JavaScript is invoked from WebViewClient. To catch the result of JavaScript functions, a WebChromeClient is set to WebView, from this listener this class will get the following values which are going to be displayed as result of test using CustomDialog with “OK” button, and also update the result to device internal database and test result also published to ViaSat server using ServerAccess class. If user clicks on “OK”, VideoLoadTestActivity will be finishes and navigates the user to TestMenuActivity. If VideoLoadTestActivity is called under the RunAllTests case, After successful test completion, CustomDialog with result will shown for 1.5 sec, After 1.5 sec CustomDialog will be dismisses, VideoLoadTestActivity will be finishes and calls TestResultActivity class.

JavaScript invoke calls:

wvVideoLoadTest.loadUrl("<javascript:alert(getInitialQuality>())");

wvVideoLoadTest.loadUrl("<javascript:alert(getInitialBufferTime>())");

wvVideoLoadTest.loadUrl("<javascript:alert(getVideoId>())");

Result Data:

* Initial Video Quality
* Initial Buffer Time
* Video ID

**TestResultActivity:**

Here in this class, the results of the Speed Test, WebPage Load Timer, Send Byte Count Measurement, Throughput and Video Load Timer tests of ViaSat will be shown by getting from local database, in table format along with others providers result for compression. Here in this class following Test Features are shown in along with values.

* DL speed
* UL speed
* Latency
* Webpage avg. load time
* Video playback buffer time
* Bandwidth

If TestResultActivity is called under the RunAllTests case, above Test Features values will be saved in local database as one record for data comparison. Here in this class it will saves latest 10 records only.

There is a button with texture “Results Database”, this class will calls TestDataCompareActivity class when user clicks on this button.

**TestDataCompareActivity:**

There are three different buttons “Dumm 1”, “Dumm 2” and “ViaSat” to show each providers data on their respective tab click. This class displays the average of following mentioned each Test Feature in table format, and it also displays the saved test result records based on the test performed date using a ListView and TestDataAdapter adapter.

* DL speed
* UL speed
* Latency
* Webpage avg. load time
* Video playback buffer time

**2. com.viasat.etrac.adapters:**

This package of eTRAC Application contains the adapter classes which are used to show list items. Here in this package following classes are there,

TestMenuAdapter

UrlLoadAdapter

TestDataAdapter

**TestMenuAdapter:**

This adapter is used in TestMenuActivity class to show the list of various test cases. Here in this class each list item view is created from getView(**int** position, View convertView, ViewGroup parent) method, here each list item view will have a textview to bind the test case name, imageview to bind the test case icon and a imageview to bind right arrow. From this class test cases will be initiated when user clicks a listitem cell by invoking ItemClickListener classes itemClicked(Object object) method.

**UrlLoadAdapter:**

This adapter is used in WebPageLoadTimerActivity class to show the list of urls along with theire status and load timings. Here in this class each list item view is created from getView(**int** position, View convertView, ViewGroup parent) method, here each list item view will have a textview to bind the url, imageview to bind url load status and a textview to bind url load time. From this class each url will be starts loading using loadViews(**int** posn) method and changes the load status using refreshView(**int** posn , **long** val) method, if all urls are loaded then from this method loadingCompleted(Vector<LinkDataDo> vecLinkDataDos) method of WebPageLoadTimerActivity class will be called to intemate that there are no urls to load.

**TestDataAdapter:**

This adapter is used in TestDataCompareActivity class to show the list of test result records which are stored in the local database. Here in this class each list item view is created from getView(**int** position, View convertView, ViewGroup parent) method, here each list item view will shows each Test Feature along with its respective value along with tested date and service provider name of each saved test record.

**3. com.viasat.etrac.common:**

This package of eTRAC Application contains commonly used classes among different classes. Here in this package following class is present,

ShareConstants

**ShareConstants:**

This class is used for maintaining common and unique keys for storing data in to SharedPreferences and to get data from SharedPreferences.

**4. com.viasat.etrac.controls:**

This package of eTRAC Application contains some customized controls. Here in this package following classes are there,

CustomDialog

LoaderDialog

ThroughputTest

**CustomDialog:**

This class extends Dialog, by using this class this application showing different alert message to user depending on situation.

**LoaderDialog:**

This class extends **Dialog**, by using this class this application showing different loader message to user depending on situation.

**ThroughputTest:**  
This class extends AsyncTask, and it is used in ThroughputActivity to perform Throughput test. When this class is called, it passes messages to ThroughputActivity using StatusListener. These listener calls can be called from different method of these calls.

OnPreExecute() --- testStarted()

onProgressUpdate(String... strings) --- testStatusUpdated(strings[0])

onCancelled() --- testCanceled()

onPostExecute(String result) --- testCompleted()

By using these messages there in the activity some statements are going execute.

In this class the main method is doInBackground(Void... voids), in this method the main throughput test will be done. Here in this method the result will passed to the ThroughputActivity by calling onProgressUpdate(String... strings) using publishProgress(output.toString()) method in doInBackground(Void... voids).

**5. com.viasat.etrac.dataobj:**

This package of eTRAC Application contains different User-Defined Data types. Here in this package following classes are there,

RouterDo

ServerInfoDo

TestResultDo

LinkDataDo

WebLoadDo

**RouterDo:**

This class is used to store and display the router and service provider details whenever needed in this application.

**ServerInfoDo:**

This class is used to store and uses the server details which is obtained in ThroughputActivity whenever needed in this application.

**TestResultDo:**

This class is used to store and display the test results in TestDataCompareActivity and TestResultActivity classes whenever needed in this application.

**LinkDataDo:**

This class is used to store and use the link name, load time, load status which is obtained in WebPageLoadTimerActivity whenever needed in this application. And also uses it in WebLoadDo class.

**WebLoadDo:**

This class is used to store and use the loaded urls number, timeout and list of LinkDataDo which is obtained in **WebPageLoadTimerActivity** whenever needed in this application.

**6. com.viasat.etrac.listeners:**

This package of eTRAC Application contains different interfaces which are used as listeners to pass messages from one class to other. Here in this package following interfaces are there,

DataListener

itemClicked

StatusListener

**DataListener:**

This interface is used to pass the downloaded data from GetWebLoadData AsyncTask to WebPageLoadTimerActivity class. It is having dataDownloaded(Object object) method in it.

**ItemClicked:**

This interface is used to pass the selected list cell positin from TestMenuAdapter class to TestMenuActivity class. It is having itemClicked(Object object) method in it.

**StatusListener:**

This interface is used to pass the different states of test from ThroughputTest AsyncTask to ThroughputActivity class. It is having testStarted(), testStatusUpdated(String data), testCompleted() and testCanceled() methods in it.

**7. com.viasat.etrac.services:**

This package of eTRAC Application contains classes which are used to call the webservices and to handle the response. Here in this package following classes are there,

GetWebLoadData

ServerAccess

**GetWebLoadData:**

This is a AsyncTask class, this class will calls the service by passed parameter using DefaultHttpClient and HttpGet classes and parses the response json string into WebLoadDo object and passes the result WebLoadDo object to WebPageLoadTimerActivity class by calling dataDownloaded(Object object) method of DataListener interface.

**ServerAccess:**

This is a AsyncTask class, this class will calls the service by passed parameter using DefaultHttpClient and HttpPost classes to published the test results at ViaSat server .

**8. com.viasat.etrac.utils:**

This package of eTRAC Application contains classes which are used across the application classes. Here in this package following classes are there,

CalenderUtils

eTracUtils

LocationUtils

LogUtils

NetworkUtils

**CalenderUtils:**

This class is used to get the Calendar and Date related data in the application. It is used in TestResultActivity class to get the current time stamp by calling getCurrentTimeStamp() method.

**eTracUtils:**

This class is used to store the app required data like Links of different test and to store and get the required data by using setter and getter methods of this class.

**LocationUtils:**

This class is used to get the location by using available location provider with LocationManager Service.

**LogUtils:**

This class is used to print the logs on logcat.

**NetworkUtils:**

This class is used to get the Network Avaialbility, Router and Service Provider Information using ConnectivityManager,WifiManager,DefaultHttpClient and HttpGet classes.