**Application Name:** Radio Ad Analyzer

**Language:** C#

***Challenge:***  *How can you analyze 24 hours Radio transmission recording in 4 hours?*

**Mechanism:**

Multi-threading.

**Brief Explanation:**

Our project is about Radio Ad checking system. In which we will check how many times a specific Ad broadcasted in 24 hours. Our basic focus would be to analyze 24 hours recording in 4 hours. Our application will give functionality of manual checking of Ads.

In the layout shown below, buttons have been provided to select the 24 hour audio which has to be analyzed in 4 hours. Ads can be added which could be used in the matching algorithm to count the number of occurrence in the audio file. More than one ads can be added to the whole process. 6 streams run concurrently and each stream has its own count of ads. When a match is found, it prompts the operator so that the operator would click the radio button to cross-check the match and confirm it.

Text fields will act as a counter for Ads. Every time Ad appears on audio file, counter will be added by 1. Again, using text fields for gross total of all Ad’s appeared on 6 channels. Date Time Picker is used to select date. Track bar is used to show for volume adjustment in manual verification of Ads.

We will create 6 threads which will be responsible to play the 24 hour audio files from 6 different locations. These locations would be calculated on the basis of the starting points of 6 equally divided chunks. Basically, we will not divide the audio file into chunks which will take quite a long time rather, we will play the audio file from 6 different locations concurrently. Ads will get compared by some comparison algorithm. By using Multi-Threading Concept these threads we will run these threads concurrently. In this way the processing time of 24 hours recording would be 4 hours.

**Library:**

**Using Sytem.Threading;** is a library that is for Multi-Threading Processes. It allows creating and accessing individual threads in a multithreaded application.

**Prototype:**

