SCHEDULING AVSAT

Broadcast Electronics Tech Note

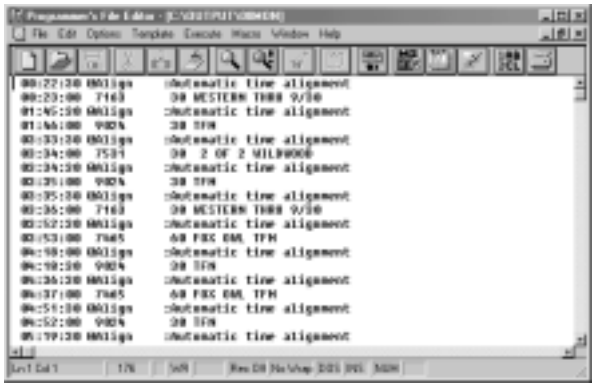
THE INFORMATION IN THIS ARTICLE APPLIES TO:

All AVSat versions

SUMMARY

AVSat reads the list of commercials that need to be played from an AudioVAULT **playlist**. Playlists can be imported from text files either generated by your traffic program or modified by a Broadcast Electronics **filter**.

MORE INFORMATION



An example of a text file in BEI format

The text file that is imported into the AudioVAULT playlist needs to be in a specific **format**, meaning it has to look a specific way in a text editor, with specific information in certain columns. Two common AudioVAULT formats are **BEI** and **Import/Export**.

The file you import into the AudioVAULT also must contain instructions pertaining to AVSat operation. For instance, AVSat loads the next day's playlist using a **Join** command. The **Join** command tells AVSat the name of tomorrow's playlist and when it needs to be loaded. The AudioVAULT playlist also defines how AVSat deals with breaks not filled by traffic, and when to load custom liners and IDs called **announcer stacks**.

Commercials are divided into commercial breaks using the playlist **auto-start** command. All AVSat needs is the contact closure to start the first commercial in a break, and each subsequent auto-started commercial plays automatically.

If your traffic program is flexible enough to write a file in an AudioVAULT format, and include these elements…

- **Auto-start commands**
- **Align statements**
- **Join commands**
- **Empty Break statements**
- **Announcer Stack load commands**

…all you need to do is import the file created by the traffic program into an AudioVAULT playlist.

If your traffic program cannot add those elements, they can be added to the file using a special process called **filtering**. Not only does the **filter program** convert the traffic file into an AudioVAULT format, it inserts those AudioVAULT-specific elements.

The procedure to import information from a Traffic System into an AudioVAULT is similar from site to site:

- The log is created in the Traffic Program.
- The log is written to a text file on a Floppy Disk or Network Drive.
- The text file is run through a BE provided **Filter** program if necessary.
- The resulting file is imported into an AudioVAULT screen and saved as a Playlist.

The Filter looks for specific information in specific columns in the Traffic output file, so the format of that file is critical. For example, our CBSI filter requires that traffic output files have this format:

Columns 6-13 must contain the Estimated Start Time in HH:MM:SS format.
Columns 15-19 must contain the AudioVAULT filename.
Columns 39-68 are comment fields.

Example of CBSI line:

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123
        1         2         3         4         5         6         7
   |05:19:00|     |0000|11111-11    |COMMERCIAL0                    |0030
```

In addition to the internal structure of the file, there are specific requirements for the CBSI filename. The extension of a CBSI log file should be the three-letter abbreviation of the day. Also, the filter uses the **first two** characters of a CBSI log file to identify the station the log is associated with. We call those characters the **station ID**. Especially if you have multiple stations, those characters need to be unique. For example:

AM-KLM.FRI	AM is the station ID, FRI is the day
AMCBS.THU	AM is the station ID, THU is the day
FM-TKP.MON	FM is the station ID, MON is the day
FM.TUE	FM is the station ID, TUE is the day
01082701.TUE	08 is the station ID, TUE is the day

The reason those first two characters need to be unique is that the files created by the filter program will have file names based on the station ID. For example:

AM-KLM.FRI	Creates an import file called AMFRI
AMCBS.THU	Creates an import file called AMTHU
FM-TKP.MON	Creates an import file called FMMON
FM.TUE	Creates an import file called FMTUE
01082701.TUE	Creates an import file called 08TUE

The station ID is also used to determine the file names specified in the **Join** command:

AM-KLM.FRI	Joins to a playlist called AMSAT
AMCBS.THU	Joins to a playlist called AMFRI
FM-TKP.MON	Joins to a playlist called FMTUE
FM.TUE	Joins to a playlist called FMWED
01082701.TUE	Joins to a playlist called 08WED

INSTALLING A FILTER PROGRAM

The **Filter** program executable is installed to a **Filter** directory, typically on a traffic computer, and will create AudioVAULT-importable traffic files, typically on a floppy in your A-Drive.

The first step in installing the filter is to find the correct one. There are several filters to chose from, each written for a specific traffic program or output format. Descriptions of the file formats required by each filter can be found in the **README** text file included with each filter. Once you've found the right filter, to install it run the **SETUP** executable. Each filter's setup routine is a bit different, but will basically ask questions including:

Where do you want to install the filter executable?
What is the station ID?
Where are the logs to be filtered located?
Do you want to include time align statements?
Where do you want the filter to write the finished file?

A description of each question asked by Setup can be found in the filter's README file.

EMPTY BREAK AND ANNOUNCER STACK FILES

Once you've installed the filter, you'll need to write the **Empty Break** and **Announcer Stack** files. In the filter's install set, you should have files called **IDDAY.BRK** and **IDDAY.STK**. Copy these and use them as your templates.

The empty break (with the **BRK** extension) files contain lists of all the breaks your station should take in a day. If traffic doesn't schedule any spots for a break, the filter knows to put in an Empty Break as a "placeholder" to absorb the satellite contact closure.

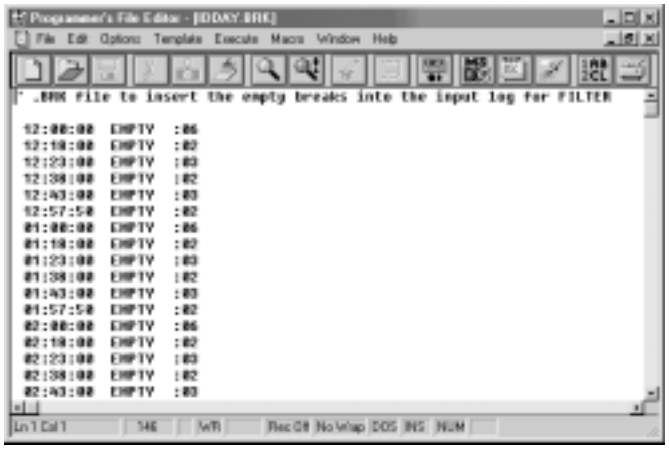
Regardless of whether the affiliate has commercials scheduled, the network will send closures at the start of each commercial break. In this example, each hour the network sends three closures: at 20, 40 and 50 minutes past the hour. These closures will start the break loaded in AVSat. If our schedule looks like this:

```
10:20:00   8070 Commercial
         + 4825 Commercial
         + 7632 Commercial
10:50:00   9030 Commercial
         + 0004 Commercial
         + 2084 Commercial
```

The closure at 20 will fire the break scheduled for 10:20:00. The next break that will be queued up is the break scheduled for 10:50:00, but the next closure will come at **40**. The end result is that the commercials scheduled for 50 will play at 40, and then your schedule is off. To get around this, the filter adds **Empty Break** statements that can compensate for unfilled commercial breaks.

```
10:20:00   8070 Commercial
         + 4825 Commercial
         + 7632 Commercial
10:40:00        EMPTY BREAK
10:50:00   9030 Commercial
         + 0004 Commercial
         + 2084 Commercial
```

The closure at 20 will fire the break scheduled for 10:20:00. The next event that will be queued up is the **EMPTY BREAK** statement. When the next closure will comes at **40** it fires the statement, which does not affect audio and does not play any commercials. The playlist then advances and gets the 10:50:00 break ready for the 50 after closure.



An example empty break file

The empty break file lists each closure that will be received.

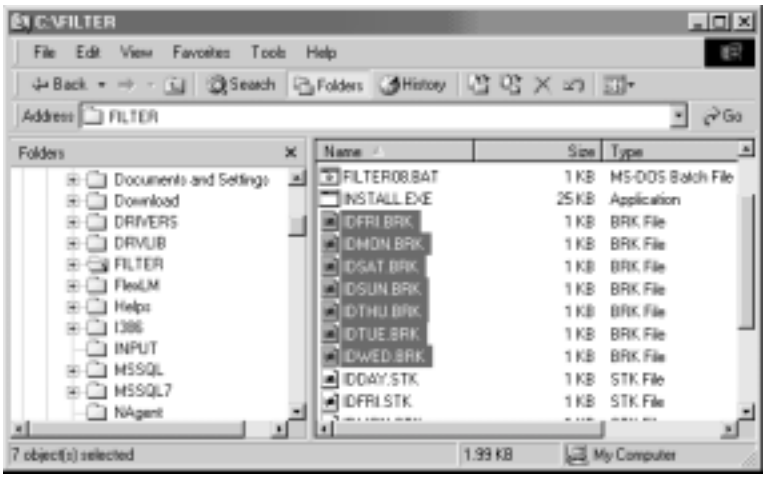
All that is required is the time of the break in 24-hour format.

Other fields in the file are for your reference. In the example to the left, the text EMPTY is included, as well as the break's duration.

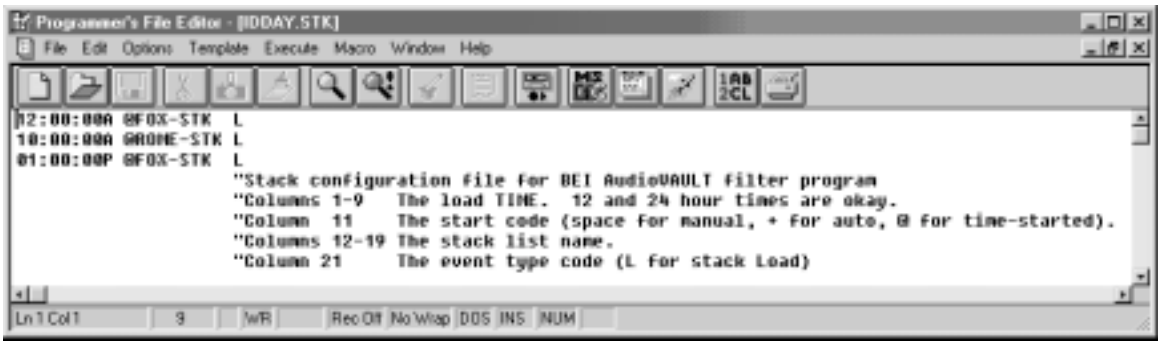
If your break times are consistent hour to hour, instead of listing each break individually, you can use **wild cards**. A station using the same clock every hour may have a break file like this:

```
" .BRK file to insert the empty breaks into the input log for FILTER

  *:20:00  EMPTY  :03
  *:40:00  EMPTY  :03
  *:50:00  EMPTY  :03

"  Column 1 is the comment flag.
"  Column 2-9 is the time field in 24-hour time.
"  Column 10-rest are reference fields.
```

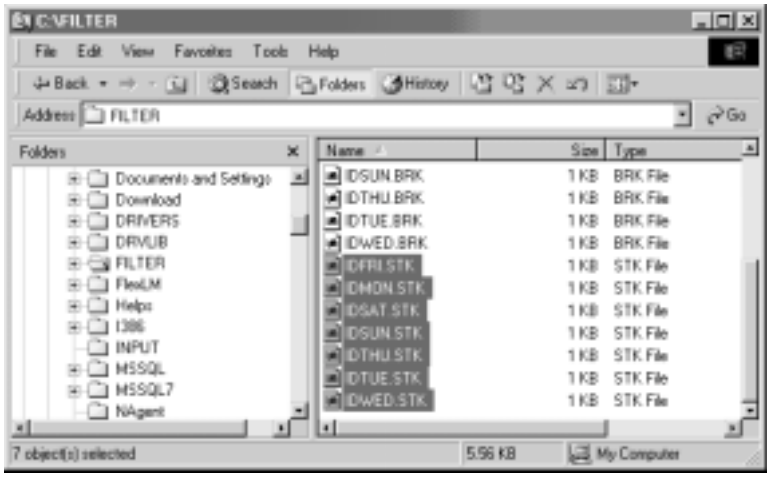
You should have seven break files stored in the filter directory with the station ID and day of the week in the file name.



The announcer stack (with the **STK** extension) files are simply a list of each Announcer Stack file you use and when they should load. These commands will be added as time-started commands at the end of the daily playlist.

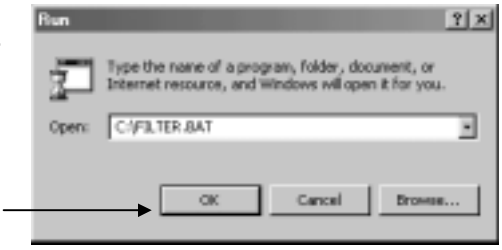


You should have seven stack files stored in the filter directory with the station ID and day of the week in the file name.



RUNNING THE FILTER

Once you've run **Setup**, and the **BRK** and **STK** files are done, the filter is installed and configured. Filtering a log is as easy as running the **FILTER.BAT** file.



Part of the setup program put the **FILTER.BAT** file on your c-drive. To run the filter, click on the Windows **Start** button, and select **Run**. **Browse** to the **FILTER.BAT** file, and double left-click on it. Then click on **OK** to filter your traffic log.

Since we specified where the files were located and where the finished file should go, we won't have to select any files. The filter will process all valid files located in the source directory. As a matter of fact, the window should pop up so quickly, you won't have time to read it!

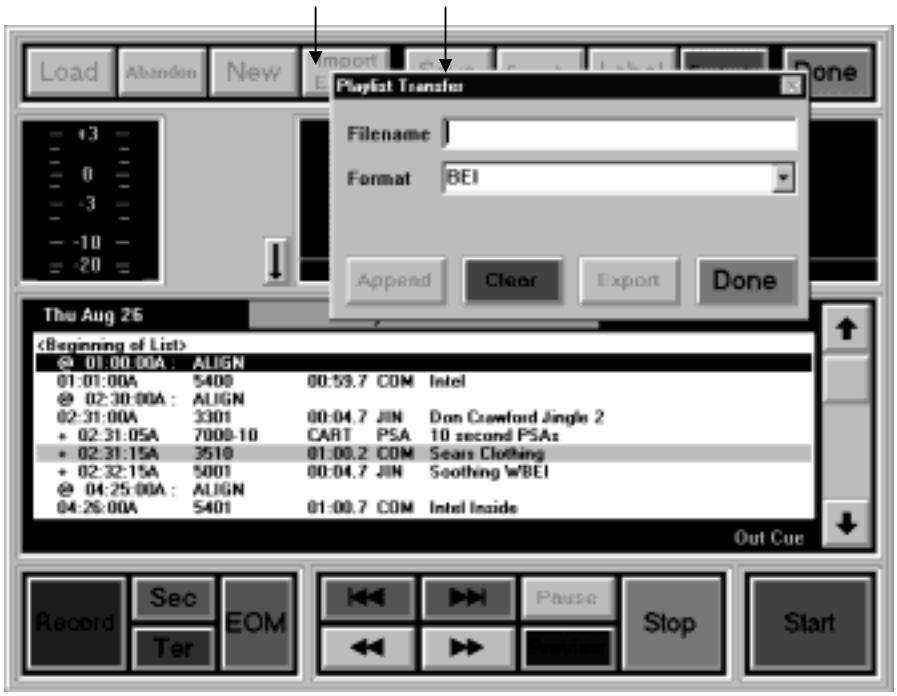
IMPORTING THE FILTERED FILE

Now that we have a filtered file in BEI format, and containing AudioVAULT-specific elements like **auto-starts** and **empty break** statements, we need to transfer that file into an AudioVAULT playlist. You should have seven daily playlists for each AVSat station you're running, one for each day of the week. The names of these playlists should match the names in the **Join** command written by the filter program. Once you've created the daily playlists, we will continually overwrite the information in them. We'll overwrite last Monday's playlist with

the new traffic information for next Monday, for instance. As long as we have filtered traffic files, we could conceivably schedule AVSat a week in advance.

First, open AVRPS. Click **Load** and type in the filename of the daily log to be overwritten with new information (**AM-TUE** for this example).

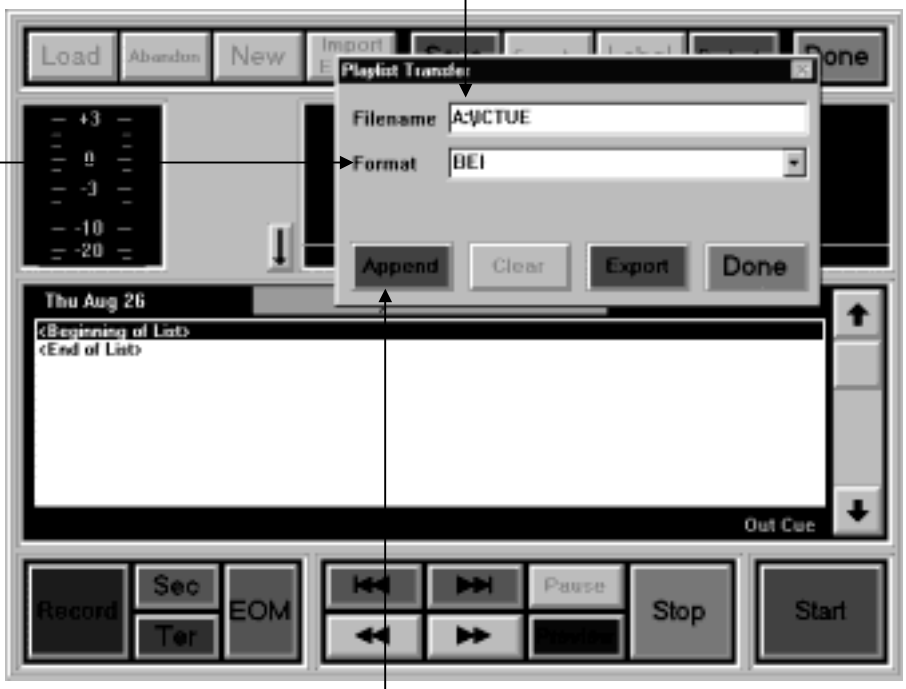
Click **List**, and then **Import/Export**. The **Playlist Transfer** dialog will appear.



Click **Clear**. The information currently in the daily log (the information that ran LAST Tuesday) will be cleared out.

Type the name of the file you want to import in the **Filename** field. This should be the path and filename of the file. In this example, we're using a file called ICTUE stored on the floppy disk in this computer's A-Drive.

Format should be BEI.



Click **Append**.

The new information from the Filtered file will be imported into the playlist.

Click **Save** to save the changes to the playlist.

The last step is to clear out this playlist. Click **Load**, hit the **backspace** key to clear out the field, and hit **Enter**. This loads a blank list (nothing), and the screen is ready for the next person.