2023

Multilarm v1.2023

DOCUMENTATION

ALI MUHAMMAD

Tuesday, December 5, 2023

Multilarm v1.2023

Documentation	Downloads	Configuration	Contact

Table of Contents

Disclaimer / License

Back to Table of contents

Introduction

Multilarm is a console application for Windows / Linux / MacOS, which plays specified audio files at specified times recursive every day, week, month or throughout the year. Its can be utilized to play Adhan at specified times using the yearly calendar, playing periodic audios in stand-alone public address systems, etc. Equipment and electricity overhead can be minimized by installing and configuring the software in the headless mode in a small single-board computer like Raspberry Pi utilizing its Linux compatibility (and of course sourcing appropriate audio HAT and speaker). The application behaviour is heavily customisable, raising wide possibilities.

Back to Table of contents

Features

1. Play any audio (.MP3 or .WAV) file or a combination of files at (same or different) specified times every day through the year
2. Play an ambient sound for a specific duration at a specified interval in between (1)
3. Display a specified text on the console associated with a particular specified time
4. Play the console text as audio in user supplied voice recordings
5. Multi-platform support: Windows, Linux, MacOS

Back to Table of contents

Installation

Windows

Please use the MSI package. This is intended for a 64-bit system. Required .NET framework and other dependencies are included in the package. You would need administrator's privileges as the installer needs access to system folder to install audio drivers. While the application size is small, added libraries of Adhan, bleep, ambient and text-to-speech sound makes the MSI package much larger. You can also download the executable and resource file separately from the Binaries folder (link below). Please ensure in that case to extract zipped resource to application root directory and to copy the bass.dll file to \windows\system32 folder. The PDB file included in the binaries are not essential for runtime, but helps with pinpointing the bug in the code, and therefore, should be copied in the application root directory.

Back to Table of contents

Linux

Tested on Raspberry Pi with framework based on Debian Bullseye. The Binaries folder contains the Linux executables, required driver and the audio library of Adhan, bleep, ambient and text-to-speech sound are contained in the zipped resource file. You would require to install drivers in appropriate locations, copy the Linux executable, configuration file and audio library in the same location and download and install required .NET framework files from Linux repos. The PDB file included in the binaries are not essential for runtime, but helps with pinpointing the bug in the code, and therefore, should be copied in the application root directory.

There is a install script for Linux hosted at our site, which can be accessed at: https://bit.ly/multilarm-linux This will automatically try to install 64-bit audio drivers for a x86_64 system and 32-bit drivers for the rest.

Please run this command on terminal for automatic installation:

```
sudo curl -s -L https://bit.ly/multilarm-linux | bash
```

This also creates a service ("multilarm.service") to boot Multilarm at system boot.

To stop Multilarm, run:

```
sudo systemctl stop multilarm.service
```

```
sudo systemctl daemon-reload
```

To permanently remove the service, in addition to stopping, please run:

```
sudo systemctl disable multilarm.service
sudo rm /etc/systemd/system/multilarm.service
```

```
sudo systemctl daemon-reload
```

Back to Table of contents

MacOS

Please copy the files from the Binaries folder along with the unzipped resource file. The PDB file included in the binaries are not essential for runtime, but helps with pinpointing the bug in the code, and therefore, should be copied in the application root directory. I will further update installation instruction when I can test this on a MacOS system (Multilarm was not tested on a MacOS system).

Back to Table of contents

Configuration

Configuration file

Configuration settings are saved at Multilarm.config.xml file, which should be located at the program root directory. The program loads configuration settings at its boot and in case of a missing tag or invalid entries for True/False values, would overwriting the values in the file. You yourself are responsible for backing up any existing configuration at risk of being overwritten in case of a missing tag or corrupt values. In case of a missing configuration file, the application will create one with default settings stored inside the application. Therefore, the application must have write-access to its root directory and if you need to reset the configuration to default, just delete the existing file. The default settings configure the application to play Adhan and relevant audios throughout the day and throughout the year according to UK (Sheffield) data and use default settings for ambient sound (for 10 seconds every 3 minutes).

Back to Table of contents

Configuration settings

The XML tags are case-sensitive. Tags are in <Tag>Value</Tag> format. It is advised not to use line feed as delimiter / within any field (except in TextData (18)), as this is parsed differently in Windows, MacOS and Linux. The following XML tags are utilized with the explanation as follows (many of the tags are interlinked, therefore, please read through the whole):

1. *RecurEveryMonth*
   Value type: True / False

   Use: Indicates whether the DateAndTimeData (16) contains only sufficient data to cover a whole month (ideally there should be 31 sets to cover the maximum days in a month). Date field (from DateIdentifier (5) to DateDelimiter (6)) in this circumstance are expected to be a number indicating the day of the month. For every month, the same data is recycled and the date number is considered as the day of that month. Only one of RecurEveryMonth (1), RecurEveryWeek (2) or RecurEveryDay (3) are supposed to be true at one time. If more than one of these are true, they are considered as (1) before (2) before (3). If none of them are true, DateAndTimeData (16) is expected to contain the data for the whole year with the date field (from DateIdentifier (5) to DateDelimiter (6)) expected to contain date and month data with DateDelimiter (6) in between.

   Default value: False

2. *RecurEveryWeek*
   Value type: True / False

   Use: Indicates whether the DateAndTimeData (16) contains only sufficient data to cover a whole week. Date field (from DateIdentifier (5) to DateDelimiter (6)) in this circumstance are expected to be first three letters of the days of the week (i.e. mon, tue, wed, thu, fri, sat, sun – case-insensitive), although can be full form as well. For every week, the same data is recycled and the date text is used to identify the day of that week. Only one of RecurEveryMonth (1), RecurEveryWeek (2) or RecurEveryDay (3) are supposed to be true at one time. If more than one of these are true, they are considered as (1) before (2) before (3). If none of them are true, DateAndTimeData (16) is expected to contain the data for the whole year with the date field (from DateIdentifier (5) to DateDelimiter (6)) expected to contain date and month data with DateDelimiter (6) in between.

   Default value: False

3. *RecurEveryDay*
   Value type: True / False

   Use: Indicates whether the DateAndTimeData (16) contains only sufficient data to cover one day. There is no date field expected in this circumstance and the whole of DateAndTimeData (16) are considered indicating different alarm times for a single day. This is recycled every day. Only one of RecurEveryMonth (1), RecurEveryWeek (2) or RecurEveryDay (3) are supposed to be true at one time. If more than one of these are true, they are considered as (1) before (2) before (3). If none of them are true, DateAndTimeData (16) is expected to contain the data for the whole year with the date field (from DateIdentifier (5) to DateDelimiter (6)) expected to contain date and month data with DateDelimiter (6) in between.

Default value: False

4. *MonthFirst*
   Value type: True / False

   Use: Consider first data presented in the date field from DateIdentifier (5) to DateDelimiter (6) as month rather that day (American style: <DateIdentifier>MM<DateDelimiter>DD rather than <DateIdentifier>DD<DateDelimiter>MM)

   Default value: False

5. *DateIdentifier*
   Value type: Character(s)

   Use: Indicates the character(s) marking the beginning of the date field in DateAndTimeData (16). This is not evaluated / not necessary if RecurEveryDay (3) is set to True.

   Default value: “*”

6. *DateDelimiter*
   Value type: Character(s)

   Use: Indicates the character(s) marking the split between day and month value in the date field in DateAndTimeData (16). This is only evaluated if the DateAndTimeData (16) is considered for the whole year, i.e. RecurEveryMonth (1), RecurEveryWeek (2) and RecurEveryDay (3) are all set as False.

   Default value: “-“

7. *TimeDelimiter*
   Value type: Character(s)

   Use: Indicates the character(s) marking the split between the date field and time data and in between time fields in DateAndTimeData (16).

   Default value: “|”

8. *UKDaylightSavings*
   Value type: True / False

   Use: Apply UK daylight savings correction at the last weeks of March and October in DateAndTimeData (16). This is only evaluated if the DateAndTimeData (16) is considered for the whole year, i.e. RecurEveryMonth (1), RecurEveryWeek (2) and RecurEveryDay (3) are all set as False.

   Default value: True

9. *PlayAmbience*
   Value type: True / False

Use: Indicates whether to play a random duration (AmbienceDuration (11)) of file at AmbienceFile (10) at a specified interval (AmbienceInterval (12))

Default value: True

10. *AmbienceFile*

    Value type: String

    Use: Indicates the path with file name and extension to be played as ambient sound. A random duration (AmbienceDuration (11)) of file at AmbienceFile (10) at a specified interval (AmbienceInterval (12)) is played if PlayAmbience (9) is set to True.

    Default value: : <Application root directory>\Adhan\

11. *AmbienceDuration*

    Value type: Numbers

    Use: Indicates time duration in milliseconds to play from a random location of file at AmbienceFile (10).

    Default value: “10” (10 seconds)

12. *AmbienceInterval*

    Value type: Numbers

    Use: Indicates the time interval in milliseconds between subsequent playing of portions of file at AmbienceFile (10).

    Default value: “180” (3 minutes)

13. *AlarmPath*

    Value type: Delimited string (“|” as delimiter)

    Use: Indicates the path to folders containing alarm files. Any number of folders can be listed delimited with “|”. File at AmbienceFile (10) and any zero-length files are excluded from playing as alarm. Only the file(s) with format(s) listed in AlarmFileFormat (14) will be included for playing.

    Default value: <Application root directory>\Adhan|<Application root directory>\Bleep

14. *AlarmFileFormat*

    Value type: String

    Use: Use: Specifies the audio file formats for the alarm audio library. Only the file(s) with format(s) listed will be included for playing. Usual supported formats are MPEG (1.0, 2.0 and 2.5 layer 3 (MP3)), MP1, MP2, OGG, WAV and AIFF. To combine more than one format, please join them with “;” in between (e.g. “wav;mp3”). Wildcard (“*”) is allowed both in filename and extension (e.g. *abc.mp* searches for filenames ending with “abc” and file extensions starting with “mp”, like mp2, mp3 etc). Only the top director(ies) mentioned in the AlarmPath (13) are searched – i.e. subfolders are not included (which can be included as separate path in AlarmPath (13)).

Default value: “mp3”

15. *AlarmIndexData*

Value type: Delimited string (“|” as delimiter between groups, “,” as delimiter within a group)

Use: For a corresponding DateAndTimeData (16) specified for the current day, a random file from the folder specified by the index (1-based) from the AlarmPath (13). For example, consider “6:26|8:20|12:11|1:44|4:00|5:48” as DateAndTimeData (16), and default values for AlarmPath (13) and AlarmIndexData (15). Hereby, at 6:26, a random file from Bleep folder is played, at 8:20, a random file from Adhan folder is played and so forth.

Default value: “2|1|2|2|2|1|1|2|1|1”

16. *DateAndTimeData*

Value type: Delimited string (“|” as delimiter)

Use: DateIdentifier (5) as marker before start of date / month / day of the week, DateDelimiter (6) as divider between day and month part of the date and TimeDelimiter (7) as delimiter between date and times and between other times. For example in the default value, date field is “1-1”, indicating 1st of January. “*” here is the DateIdentifier (5) and “-” here is the DateDelimiter (6). “|” is the TimeDelimiter (7) and in between are the time fields indicating time of the day. The default data is for UK Sheffield prayer time for the year.

Default value: *1-1|6:26|8:20|12:11|1:44|4:00|5:48*2-1|6:26|8:20|12:11|1:45|4:01|5:49*…

17. *DateAndTimeDataFormatInEffect*

Value type: Delimited string (“|” as delimiter)

Use: This string effectively and dynamically modifies DateAndTimeData (16) for a particular day and renders a new set of time data for that day. The first part of the numbers is a 1-based index of the time fields for a particular day. The second part adds (+) or subtracts (-) the following numbers (in minutes) from that matching indexed data. The time fields are then considered for alarm on that date. For example, consider “6:26|8:20|12:11|1:44|4:00|5:48” as DateAndTimeData (16) for a particular day (please note that the date field is omitted and the first time field has an index of 1 – hence 1-based) and default value for DateAndTimeDataFormatInEffect (17). So the 5 minutes would be subtracted from the first time field, the second time field will be represented by the first time field, 10 minutes will be added to the second time field to be displayed as the fourth time field and so forth. So the effective time field string for that would be: “6:26-5|6:26|8:20|8:20+10|12:11-10|12:11|1:44|4:00-7|4:00|5:48” and therefore: “6:21|6:26|8:20|8:30|12:01|12:11|1:44|3:53|4:00|5:48”. After this “effective” string is formed, it will be sorted in ascending order – unless a major addition or subtraction of minutes made, it should not be causing much effect.

Default value: “1-5|1|2|2+10|3-10|3|4|5-7|5|6”

18. *TextData*

Value type: Delimited string (TextDelimiter (19) as delimiter)

Use: Contains text data to be displayed on console. The text data are index-matched with DateAndTimeData (16) and corresponding text is displayed for corresponding time field (note that initial date field is excluded from matching). For example, consider “6:26|8:20|12:11|1:44|4:00|5:48” as DateAndTimeData (16) for a particular day (please note that the date field is omitted and the first time field has an index of 1 – hence 1-based) and “A<TextDelimiter>B<TextDelimiter>C<TextDelimiter>D<TextDelimiter>E<TextDelimiter>F” for TextData (18). Hereby, at 6:26 to 8:20, A will be printed on the console, at 8:20 to 12:11, B will be printed on the console and so forth. To reference next time field or time to next time field, “#ALARM+1#” and “#TIMETOALARM+1#” can be used within the text as surrogate respectively. To skip the next field and reference the time field after that (“next after the next”), “#ALARM+2#” and “#TIMETOALARM+2#” can be used within the text as surrogate respectively. For example, from our previous DateAndTimeData (16) data for a particular day, if it is 07:00 o’clock, the following text replacement will occur from the TextData (18):

“#ALARM+1#” → 8:20

“#TIMETOALARM+1#” → 1 hour(s) and 20 minute(s)

“#ALARM+2#” → 12:11

“#TIMETOALARM+2#” → 4 hour(s) and 11 minute(s)

Default value: “It is time for Esha”…

19. *TextDelimiter*

Value type: Character(s)

Use: Separates text data into groups to be index-matched with DateAndTimeData (16) for a particular day for displaying on the console.

Default value: “|”

20. *TextToSpeech*

Value type: True / False

Use: Indicates whether written text in the console would be read aloud upon any changes in the console text (from TextData (18)) using the word audio library.

Default value: True

21. *TTSPath*

Value type: String

Use: Indicates the location of the word audio library for console text to speech operation. Only the file(s) with format(s) listed in TTSFileFormat (22) will be included for playing.

Default value: <Application root directory>\TTS

22. *TTSFileFormat*

Value type: String

Use: Specifies the audio file formats for the word audio library. Only the file(s) with format(s) listed will be included for playing. Usual supported formats are MPEG (1.0, 2.0 and 2.5 layer 3 (MP3)), MP1, MP2, OGG, WAV and AIFF. To combine more than one format, please join them with “;” in between (e.g. “wav;mp3”). Wildcard (“*”) is allowed in extension (e.g. mp* searches for file extensions starting with “mp”, like mp2, mp3 etc). Only the top directory mentioned in the TTSPath (21) are searched – i.e. subfolders are not included.

Default value: “mp3”

Back to Table of contents

Bugs

Handled and unhandled exceptions (“bugs”) are automatically logged (appended) to MultilarmError.log in the application root directory. Therefore, the program must have write access to application root directory. Please include MultilarmError.log and Multilarm.config.xml from the application root directory to report any bugs.

Back to Table of contents

Tips and Tricks

- If you have got a list of timetables (e.g. permanent Salat calendar), use Microsoft Word to properly format the date and time, removing any unwanted spaces and characters, line feeds etc. Use special character placeholders in Find/Replace box (e.g. ^p for line feed).
- It is possible in Linux / Raspberry Pi to auto-start the application at boot in a headless mode and therefore obviate the need for further interaction with the system once it is properly set up.

- To set up a word audio library to read aloud any text from the console screen, you need to first identify unique words in your text (TextData (18)). There are quite a few online tools (e.g. https://design215.com/toolbox/wordlist.php). Thereafter those words can be individually recorded with appropriate pause following each word (probably 250 milliseconds?). Alternatively, all the words can be recorded in a long audio files, which can later be clipped with and audio editing tool (e.g. Audible) and named as the word contained within.
- For text-to-speech covering any time data, you would need to provide recorded audio for word covering number 0 to 20, 30, 40 and 50 – a total of 23 files.
- Please use 29-day month for February if using with the whole year data and ensure any daylight savings adjustments are applied after March or October finished in the UK.
- Please include MultilarmError.log and Multilarm.config.xml from the application root directory to report any bugs.

Back to Table of contents

Console Key Commands

Control + T: Test play any random audio file from a random folder in AlarmPath (13) matching AlarmFileFormat (14)

Control + A: Test play AmbienceFile (10) for AmbienceDuration (11)

Control + S: Test play current display text from TTSPath (21) matching TTSFileFormat (22)

Control + X: Exit program

Back to Table of contents

Command-line Arguments

None parsed currently.

Back to Table of contents

Download Links (folder locations)

Download folder

Binaries

Windows:	Windows x64 binaries
Linux:	Linux ARM 32-bit binaries
	Linux x86 64-bit binaries
MacOS:	MacOSx 64-bit binaries
Resource file (for Adhan):	Zipped resources

Installer

Windows (MSI): WIndows x64 MSI installer

Configuration file

Default: Multilarm.config.xml (for application root)

With Bangladesh prayer times: Multilarm.config.xml (for application root)

Back to Table of contents

Contact, Feedbacks, Suggestions, Bug reports

Ali Muhammad

Email: multilarm@zayeed.org

Web Repository: zayeed.org

Github Repository: Multilarm Github

Please include MultilarmError.log and Multilarm.config.xml from the application root directory to report any bugs.

Back to Table of contents

Acknowledgement

My wife, kids

Websites hosting Adhan and Bleep mp3s

Back to Table of contents

Donations

PayPal handle: @AliMuhammadK62

Back to Table of contents