

# Playrec

## Multi-channel Matlab Audio

[Home](#)   [Getting Started](#)   **[Usage](#)**   [Documentation](#)   [Download](#)   [Contact](#)

### Usage Overview

This page provides a basic outline of how to use the Playrec utility. More information on any command can be found by using `playrec('help', 'command_name')` or simply `playrec help command_name`.

To simplify operation, all functionality is accessed through one function in Matlab. To achieve this the first argument in the function call is always the name of the command/operation required, eg `playrec('getDevices')` or `playrec('isInitialised')`. If additional arguments are required then they are specified after this, eg `playrec('init', 48000, 1, -1)`. A list of all available commands can be displayed by supplying no arguments.

Before any audio can be played or recorded, the utility must be initialised to use the required sample rate and device(s). Initialisation is achieved using the `init` command, supplying the ID of the required audio device(s) as returned by `getDevices`. Once successfully initialised, the sample rate or device(s) to be used cannot be changed without first resetting the utility using the `reset` command. This clears all previously recorded data so use it with care. To check if the utility is currently initialised, use the `isInitialised` command.

The utility divides time up into pages with no restrictions on the duration of any one page, although with very short pages skipping in the audio may occur if they cannot be supplied fast enough. There can be as many pages as required provided the utility can allocate enough memory. Pages are joined together sequentially in the order they are added, with each page starting the sample after the previous page finishes. A page can contain samples that are to be output on one or more channels and buffers to store recorded samples on one or more channels. The duration of

a page is determined by the longest channel contained within the page. Therefore if, for example, the record channels are 1000 samples long whilst output channels are only 900 samples long, the page will be 1000 samples long and the final 100 output samples of the page will automatically be set to 0.

When each page is added, the channels that are to be used for recording and/or output are specified (depending on the command used to add the page). The channels used must be within the range specified during initialisation and no channel can be duplicated within a channel list. Within these limits, the channel list for each page can be different and each list can contain as many or as few channels as required in any order. All output channels not provided with any data within a page will output 0 for the duration of the page. Similarly, during any times when there are no further pages to process 0 will be output on all channels.

Each page has a unique number which is returned by any of the commands used to add pages (`playrec`, `play` or `rec`). When a page is added, the utility does not wait until the page has completed before returning. Instead, the page is queued up and the page number can then be used to check if the page has finished, using `isFinished`. Alternatively a blocking command, `block`, can be used to wait until the page has finished. To reduce the amount of memory used, finished pages are automatically condensed whenever any command is called in the utility. If a page contains any recorded data, this is left untouched although any output data within the page is removed. If the page does not contain any recorded data, the whole page is deleted during this page condensing. For this reason if either `isFinished`, `block` or `delPage` indicate the page number is invalid this means the page either never existed or has already finished and then been deleted during page condensing.

For pages containing recorded data, the data can be accessed using the `getRec` command once the page is finished (indicating the recording has completed). This does not delete the data so it can be accessed as many times as required. To delete the recorded data, the whole page must be deleted using the `delPage` command. This command will delete pages nomatter what their current state: waiting to start, currently active or finished. If no page number is supplied, all pages will be deleted, again regardless of their state so use with care.

To ascertain which pages are still left in memory, the `getPageList` command can be used, returning a list of the pages in chronological order. NOTE: there may have been gaps of silence or other pages between consecutive pages in this list due to pages either being automatically or explicitly deleted as detailed above. To determine if there were gaps between pages due to all pages finishing processing before new ones are added, the commands `getSkippedSampleCount` and `resetSkippedSampleCount` can be used.

The page that is currently being output is returned by `getCurrentPosition`, along with an approximate sample position within the page. Additionally, the page number of the last completed page still resident in memory is returned by `getLastFinishedPage`. NOTE: this might not be the most recent page to finish if that page has been deleted either during page condensing (ie contained no recorded data) or through the use of `delPage`.

Finally, the utility can be paused and resumed using the `pause` command. This will manipulate all output and recording channels simultaneously to ensure synchronisation is always maintained. This command can also be used to ascertain if the utility is currently running or paused.

Software License

© 2008-2014 Robert Humphrey