Playrec

Multi-channel Matlab Audio

Available playrec commands

This page describes all available Playrec commands. See usage for more information on how and when to use the commands.

help	Provides usage information for each command
about	Displays information about the playrec utility
overview	Displays an overview on using this playrec utility
getDevices	Returns a list of available audio devices
init	Initialises the utility
reset	Resets the system to allow re-initialisation
isInitialised	Indicates if the system is initialised
playrec	Adds a new page with simultaneous input and output
play	Adds a new output only page
rec	Adds a new input only page
pause	Sets or queries the current pause state
block	Waits for the specified page to finish before returning
isFinished	Indicates if the specified page has finished
getRec	Returns the samples recorded in a page
delPage	Deletes the specified page or all pages
getCurrentPosition	Returns the currently active page and sample number
getLastFinishedPage	Returns the page number of the last completed page
getPageList	Returns an ordered list of all page numbers

getFramesPerBuffer	Returns internal number of frames per buffer
getSampleRate	Returns the current sample rate
getStreamStartTime	Returns the time at which the stream was started
getPlayDevice	Returns the current output (play) device
getPlayMaxChannel	Returns the current maximum output (play) channel
getPlayLatency	Returns the current output (play) device latency
getRecDevice	Returns the current input (record) device
getRecMaxChannel	Returns the current maximum input (record) channel
getRecLatency	Returns the current input (record) device latency
resetSkippedSampleCount	Resets the skipped samples counter
getSkippedSampleCount	Returns the number of skipped samples

help

Input Arguments:

commandName	name of the command for which information is required

Description:

Displays command specific usage instructions.

Top

about

Output Arguments:

aboutInfo	String containing information about this build of playrec. If no output argument is specified then the information is printed in the command window.

Description:

Displays information about the playrec utility

Top

overview

Output Arguments:

overviewInfo	String containing information about how to use playrec. If no output argument is specified then the information is printed in the command window.

Description:

Displays an overview on using this playrec utility

Top

getDevices

Output Arguments:

deviceList Structure array containing the following fields for each device:
'deviceID' - ID used to refer to the device,
'name' - textual name of the device,
'hostAPI' - the host API used to access the device,
'defaultLowInputLatency' - device default input latency used for interactive performance. This is the value suggested to the soundcard when the device is used for input.
'defaultLowOutputLatency' - device default output latency used for interactive performance. This is the value suggested to the soundcard when the device is used for output.
'defaultHighInputLatency' - device default input latency for robust non-interactive applications (eg. playing sound files),
'defaultHighOutputLatency' - device default output latency for robust non-interactive applications (eg. playing sound files),
'defaultSampleRate' - device default sample rate,
'inputChans' - maximum number of input channels supported by the device
'outputChans' - maximum number of output channels supported by the device

Description:

Returns information on the available devices within the system, including ID, name, host API and number of channels supported.

Top

init

Input Arguments:

sampleRate
the sample rate at which both devices will operate

playDevice
the ID of the device to be used for sample output (as returned by 'getDevices'), or -1 for no device (ie output not required)

recDevice
the ID of the device to be used for sample input (as returned by 'getDevices'), or -1 for no device (ie input not required)

playMaxChannel {optional}
a number greater than or equal to the maximum channel that will be used for output. This must be less than or equal to the maximum number of output channels that the device supports. The value is ignored if playDevice is -1.

recMaxChannel {optional}
a number greater than or equal to the maximum channel that will be used for input. This must be less than or equal to the maximum number of input channels that the device supports. The value is ignored if recDevice is -1.

framesPerBuffer {optional}
the number of samples to be processed in each callback within the utility (ie the length of each block of samples transferred between the utility and the soundcard). The lower the value specified the shorter the latency but also the greater the likelihood of glitches within the audio. This has no influence on the size of pages that can be used. The default is 0 which lets the utility use an optimal, and potentially different, value in each callback. A value other than the default may introduce a second layer of buffering, increasing latency, and so should only be used in exceptional circumstances.

playSuggestedLatency {optional}
the play latency, in seconds, the device should try to use where possible. Defaults to the default low output latency for the device.

recSuggestedLatency {optional}
the record latency, in seconds, the device should try to use where possible. Defaults to the default low input latency for the device.

Description:

Configures the utility for audio input and/or output based on the specified configuration. If successful the chosen device(s) will be running in the background waiting for the first pages to be received. If unsuccessful an error will be generated containing an error number and description.

All channel numbers are assumed to start at 1. The maximum number of channels support by the device will be used if the maximum channel number is not specified. Specifying a maximum number of channels verifies that the device will support them and slightly reduces the utility's processor usage.

If an optional value is specified, all previous optional values must also be specified.

Top

reset

Description:

Resets the system to its state prior to initialisation through the 'init' command. This includes deleting all pages and stopping the connection to the previously selected audio device(s). Generates an error if the utility is not already initialised - use 'isInitialised' to determine if the utility is initialised.

Use with care as there is no way to recover previously recorded data once this has been called.

Top

isInitialised

Output Arguments:

currentState 1 if the utility is currently initialised, otherwise 0.

Description:

Indicates if the system is currently initialised, and hence if 'reset' or 'init' can be called without generating an error.

Top

playrec

Input Arguments:

playBuffer a MxN matrix containing the samples to be output. M is the number of samples and N is the number of channels of data.

playChanList a 1xN vector containing the channels on which the playBuffer samples should be output. N is the number of channels of data, and should be the same as playBuffer (a warning is generated if they are different, but the utility will still try and create the page). Can only contain each channel number once, but the channel order is not important and does not need to include all the channels the device supports. All output channels no specified will automatically output zeros. The maximum channel number cannot be greater than that specified during initialisation.

recDuration the number of samples that should be recorded in this page, or -1 to record the same number of samples as in playBuffer.

recChanList a row vector containing the channel numbers of all channels to be recorded. Can only contain each channel number once, but the channel order is not important and does not need to include all the channels the device supports. This order of channels is used when recorded samples are returned by 'getRec'. The maximum channel number cannot be greater than that specified during initialisation.

Output Arguments:

pageNumber a unique integer number identifying the page that has been added - use this with all other functions that query specific pages, such as 'isFinished'.

Description:

Adds a new page containing both sample input (recording) and output (playing). Generates an error if the required memory cannot be allocated or if any other problems are encountered.

The length of the page is equal to whichever is longer: the number of samples to play or the number of samples to record.

Top

play

Input Arguments:

playBuffer a MxN matrix containing the samples to be output. M is the number of samples and N is the number of channels of data.

playChanList a 1xN vector containing the channels on which the playBuffer samples should be output. N is the number of channels of data, and should be the same as playBuffer (a warning is generated if they are different, but the utility will still try and create the page). Can only contain each channel number once, but the channel order is not important and does not need to include all the channels the device supports. All output channels no specified will automatically output zeros. The maximum channel number cannot be greater than that specified during initialisation.

Output Arguments:

pageNumber a unique integer number identifying the page that has been added - use this with all other functions that query specific pages, such as 'isFinished'.

Description:

Adds a new page containing only sample output (playing). Generates an error if the required memory cannot be allocated or if any other problems are encountered.

The page is the same length as that of playBuffer.

Top

rec

Input Arguments:

recDuration the number of samples that should be recorded on each channel specified in recChanList.

recChanList a row vector containing the channel numbers of all channels to be recorded. Can only contain each channel number once, but the channel order is not important and does not need to include all the channels the device supports. This order of channels is used when recorded samples are returned by 'getRec'. The maximum channel number cannot be greater than that specified during initialisation.

Output Arguments:

pageNumber a unique integer number identifying the page that has been added - use this with all other functions that query specific pages, such as 'isFinished'.

Description:

Adds a new page containing only sample input (recording). Generates an error if the required memory cannot be allocated or if any other problems are encountered.

The page is recDuration samples long.

Top

pause

Input Arguments:

newPause {optional} the new state of the utility: 1 to pause or 0 to resume the stream. This can be either a scalar or logical value. If newState is the same as the current state of the utility, no change occurs.

Output Arguments:

currentState	the state of the utility (including the update to newPause if newPause is specified): 1 if the utility is paused or otherwise 0.

Description:

Queries or updates the current pause state of the utility. If no argument is supplied then just returns the current pause status, otherwise returns the status after applying the change to newPause.

Top

block

Input Arguments:

pageNumber {optional}	the number of the page to wait until finished

Output Arguments:

completionState	1 if either pageNumber is a valid page and has finished being processed or pageNumber was not specified and all pages have finished being processed. Note that page validity refers to when the function was called and so now the page has finished it may no longer be a valid page due to automatic page condensing. -1 if the specified page is invalid or no longer exists. This includes pages that have automatically been condensed, and hence have

finished.

0 if the stream is currently paused and neither return values of 1 or -1 apply.

Description:

Waits for the specified page to finish or, if no pageNumber is supplied, waits until all pages have finish. Note that the command returns immediately if the utility is paused to avoid the system locking up.

This uses very little processing power whilst waiting for the page to finish, although as a result will not necessarily return as soon as the page specified finishes. For a faster response to pages finishing use the 'isFinished' command in a tight while loop within MATLAB, such as

```
while(playrec('isFinished', pageNumber) == 0);end;
```

This will run the processor at full power and will be very wasteful, but it does reduce the delay between a page finishing and the MATLAB code continuing, which is essential when trying to achieve very low latency.

Top

isFinished

Input Arguments:

pageNumber {optional}	the number of the page being tested

Output Arguments:

completionState 1 if either pageNumber is a valid page that has finished being processed or pageNumber was not specified and all pages have finished being processed.

-1 if the specified page is invalid or no longer exists. This includes pages that have automatically been condensed, and hence have finished.

0 if either pageNumber is a valid page that has not finished being processed or pageNumber was not specified and not all pages have finished being processed.

Description:

Indicates if the specified page is finished or, if no pageNumber is supplied, indicates if all pages have finished.

Top

getRec

Input Arguments:

pageNumber used to identifying the page containing the required recorded data

Output Arguments:

recBuffer a MxN matrix where M is the number of samples that have been recorded

	and N is the number of channels of data
recChanList	a 1xN vector containing the channel numbers associated with each channel in recBuffer. These channels are in the same order as that specified when the page was added.

Description:

Returns all the recorded data available for the page identified by pageNumber. If the page specified does not exist, was not specified to record any data, or has not yet started to record any data then empty array(s) are returned. If the page is currently being processed, only the recorded data currently available is returned.

Top

delPage

Input Arguments:

pageNumber {optional}	the number of the page to be deleted.

Output Arguments:

completionState	0 if nothing is deleted (either there are no pages in the page list or, if pageNumber was specified, no page with the specified number exists), otherwise 1 is returned.

Description:

Deletes either the specified page or, if no pageNumber is supplied, deletes all pages. Pages can be in any state when they are deleted - the do not have to be finished and they can even be deleted part way through being processed without any problems (in this case the utility will automatically continue with the next page in the page list).

Top

getCurrentPosition

Output Arguments:

currentPage	the current page number, or -1 if either the utility is not initialised or no page is currently being processed (there are no pages in the list or all pages are finished).
currentSample	the current sample number within currentPage, or -1 if currentPage is also -1. This is only accurate to maxFramesPerBuffer samples, as returned by 'getFramesPerBuffer'.

Description:

Returns the sample and page number for the last sample transferred to the soundcard. Due to sample buffering this will always be slightly further through a page than the actual sample being output by the soundcard at that point in time. For pages that record input, the sample number shows how many samples have been recorded by the page, up to the recording length limit of the page.

Top

getLastFinishedPage

Output Arguments:

lastPage	pageNumber of the most recently finished page still resident in memory.

Description:

Returns the page number of the last finished page still resident in memory. Due to automatic condensing/removal of pages that are no longer required, such as finished pages with only output data, this may not be the most recent page to have finished. Put another way, this returns the page number of the last finished page in the pageList returned by 'getPageList'.

Top

getPageList

Output Arguments:

pageList	a 1xN vector containing the chronological list of pages, where N is the number of pages resident in memory.

Description:

Returns a list of all the pages that are resident in memory. The list is ordered chronologically from the earliest to latest addition.

Due to automatic condensing/removal of pages that are no longer required, such as finished pages with only output data, this will not be a complete list of all pages that have ever been used with the utility.

Top

getFramesPerBuffer

Output Arguments:

suggestedFramesPerBuffer	the number of frames returned by the utility internally during each callback as specified during initialisation, or -1 if the utility is not initialised.
minFramesPerBuffer	the minimum number of frames actually processed by the utility internally during a callback, or -1 if the utility is not initialised.

maxFramesPerBuffer	the maximum number of frames actually proccessed by the utility internally during a callback, or -1 if the utility is not initialised.

Description:

Returns the number of frames (samples) that are processed by the callback internally within the utility (ie the length of each block of samples sent by the utility to the soundcard). This is either the value specified when using 'init', or the default value if the optional argument was not specified. A value of 0 means the utility is using an optimal, but potentially varying, value.

Top

getSampleRate

Output Arguments:

suggestedSampleRate	the sample rate used during initialisation or -1 if the utility is not initialised.
sampleRate	the current sample rate (obtained from the hardware if possible) or -1 if the utility is not initialised.

Description:

Returns the sample rate that was specified when using 'init'.

Top

getStreamStartTime

Output Arguments:

streamStartTime	time at which the stream was started (in seconds since the Epoch), or -1 if the utility is not initialised.

Description:

Returns the unix time when the stream was started (number of seconds since the standard epoch of 01/01/1970).

This is included so that when using the utility to run experiments it is possible to determine which tests are conducted as part of the same stream, and so identify if restarting the stream (and hence the soundcard in some scenarios) may have caused variations in results.

Top

getPlayDevice

Output Arguments:

playDevice the deviceID for the output (play) device or -1 if no device was specified during initialisation or the utility is not initialised.

Description:

Returns the deviceID (as returned by 'getDevices') for the currently selected output device.

Top

getPlayMaxChannel

Output Arguments:

playMaxChannel the maximum output (play) channel number that can currently be used, or -1 if either no play device was specified during initialisation or the utility is not initialised.

Description:

Returns the number of the maximum output (play) channel that can currently be used. This might be less than the number of channels that the device can support if a lower limit was specified during initialisation.

Top

getPlayLatency

Output Arguments:

playSuggestedLatency	the suggested latency for the output (play) device used during initialisation, or -1 if no device was specified during initialisation or the utility is not initialised.
playLatency	the actual latency for the output (play) device or -1 if no device was specified during initialisation or the utility is not initialised.

Description:

Returns the output latency for the currently selected output device as well as the suggested output latency used during initialisation

Top

getRecDevice

Output Arguments:

recDevice	the deviceID for the input (record) device or -1 if no device was specified during initialisation or the utility is not initialised.

Description:

Returns the deviceID (as returned by 'getDevices') for the currently selected input device.

Top

getRecMaxChannel

Output Arguments:

recMaxChannel	the maximum input (record) channel number that can currently be used, or -1 if either no record device was specified during initialisation or the utility is not initialised.

Description:

Returns the number of the maximum input (record) channel that can currently be used. This might be less than the number of channels that the device can support if a lower limit was specified during initialisation.

Top

getRecLatency

Output Arguments:

recSuggestedLatency	the suggested latency for the input (record) device used during initialisation, or -1 if no device was specified during initialisation or the utility is not initialised.
recLatency	the actual latency for the input (record) device or -1 if no device was specified during initialisation or the utility is not initialised.

Description:

Returns the input latency for the currently selected input device as well as the suggested input latency used during initialisation

Top

resetSkippedSampleCount

Description:

Resets the counter containing the number of samples that have been 'missed' due to no new pages existing in the page list. See the help on 'getSkippedSampleCount' for more information.

Top

getSkippedSampleCount

Output Arguments:

skippedSampleCount	the number of frames (samples per channel) transferred with the soundcard that have occurred when there are no unfinished pages in the pageList, or -1 if the utility is not initialised

Description:

Returns the counter containing the number of samples that have been 'missed' due to no new pages existing in the page list when the soundcard requires samples to be transferred. The term 'missed' is specifically referring to the case where multiple consecutive pages are used to record a continuous audio stream (and so input samples are missed), but is the same also for output samples because the input and output samples within a page are always processed simultaneously.

This value is incremented by one for every frame (ie one sample on every input/output channel) of data communicated between the utility and soundcard that occurred whilst there were no new pages in the page list. Using this it is possible to determine, from within MATLAB, if any glitches in the audio have occurred through not adding a new page to the page list before all other pages have finished, such as in the case where the code within MATLAB is trying to play/record a continuous stream.

The counter can be reset using 'resetSkippedSampleCount' so to check for any breaks in a continuous stream of pages: add the first page of the stream; reset the counter; continue to add pages as required; if getSkippedSampleCount ever returns a value greater than zero then there has been a break in the stream.

Top

Software License