

mubu

MuBu, multi-buffer container for sound description and motion capture data

Description

Multi-buffer container for sound description and motion capture data.

The MuBu multi-buffer can contain multiple tracks with an individual timing (periodically *sampled* or *time-tagged* data) that are represented temporally aligned. Essentially, a track is a contiguous array of track elements including a 2-dimensional matrix of numeric values (representing scalars, vectors or matrices using 32-bit floating-point values), as well as, optionally, a time-tag and/or an item of *extra data*. Currently, *label* is the only supported type of extra data, allowing to associate a symbol to each track element. The matrix column names of a track can be named to support a notion of semantics for the access to the data and its processing by other objects. As an option, the matrix of each track element can have a variable number of rows.

In summary, the following configuration options define the track's *kind* when it is created:

- whether the track elements are time-tagged or not (*timetagged*)
- the matrix dimensions (*matrixcols* and *matrixrows*)
- the extra data type (*extradata*)
- whether the track elements' matrices can have a variable number of rows or not (*matrixrowsvar*)

In order to keep the implementation of the MuBu track simple, the a track is preallocated to a given maximum size (capacity). The maximum size and the configuration options of a track can not be changed. Nevertheless, the *modifytrack* message allows for replacing a track with a track of another kind making a copy of the original track's content.

Many MuBu externals such as the [imubu](#) editor and the synthesis externals (such as [mubu.concat~](#)), interpret the data stored in the different tracks of a container as being temporally aligned to a sound stored in a [buffer~](#) or [polybuffer~](#).

By default a container has a single *buffer*. A container with multiple buffers repeats the same structure of tracks multiple times. Each buffer of a MuBu container may be associated (by its index) to a buffer of a [polybuffer~](#). While the content, the size, and the maximum size as well as other parameters (e.g. the sample rate) can be different for the same track in different buffers, corresponding tracks of different buffers have to be of the same *kind* (see above). Consequently, when adding a track to a container with multiple buffers, an empty track is also added to the other buffers of the given container. A track of a container with multiple buffers can only be replaced by a track of the same kind (*addtrack*, *replacetrack*, *read*, etc.). Most of the messages sent to a MuBu container or (using the *track* message) to a particular track of the container, concern only the *current* buffer of the container. The current buffer and the number of buffers can be changed by attributes or messages (*bufferindex* and *numbuffers*).

Arguments

<i>Name</i>	<i>Type</i>	<i>Opt</i>	<i>Description</i>
mubu name	symbol		Name of the MuBu container.

Messages

<i>addtrack</i>	max size [int] matrix cols [int] matrix rows [int] attributes [list]		Adds an empty track to the MuBu container or replaces an existing named track by an empty track. In addition to the arguments, the message allows for using an attribute syntax to determine the configuration options and additional parameters of the created track.
-----------------	---	--	--

The following attributes are implemented:

- @name [symbol] — track name (def: unnamed)
- @maxsize [int] — maximum size (def: 256)
- @matrixcols [int] — number of matrix columns (def: 1)
- @matrixrows [int] — number of matrix rows (def: 1)
- @matrixcolnames [list] — list of matrix column names
- @matrixrowsvar [atom: 0 | 'no' | 1 | 'yes'] — number of matrix rows is variable or not (def: 'no')
- @timetagged [atom: 0 | 'no' | 1 | 'yes'] — track is time-tagged or not (def: 'no')
- @samplerate [float] — sample rate and sample period (def: 1)
- @sampleperiod [float] — sample period and sample rate (def: 1000)
- @sampleoffset [float] — sample offset time (def: 0)
- @extradata [symbol: 'none' | 'label'] — track extra data type (def: 'none')
- @info [list] — list of track info keys and values

Attention: When a track is replaced by a new track of a different kind in a particular buffer, the corresponding tracks in the other buffers are also modified respectively.

(drag)		(internal)
deltrack	[none]	(deprecated — see removetrack)
clear	track id [atom]	Removes all tracks from the container or clear a given track. If no track id is specified, all tracks are removed from the container, otherwise the given track is cleared.
getname		Outputs the name of the MuBu container (prefixed by 'name').
getnumbuffers		Outputs the current number of buffers (prefixed by 'numbuffers').
getnumtracks		Outputs the current number of tracks (prefixed by 'numtracks').
merge	track id [atom] track id [atom]	(not yet implemented)
modifytrack	track id [list] max size [int] matrix cols [int] matrix rows [int] attributes [list]	Replaced the given track by a track with the given configuration parameters and parameters copying the content of the original track. The arguments and attributes for creating the new track (following the track id), are identical to those of the addtrack message.
(mouse)		(internal)
print		Prints a summary of the container's buffers and the current buffer's content into the Max window.
read	file path [symbol]	Imports container data from a file creating one or more new tracks. The newly created tracks may be appended to the container or replace existing tracks named tracks. If no file path is given, a file open dialog allows for choosing a file in the file system.

Similar to the messages addtrack, replacetrack, and modifytrack, the read message allows for the definition of attributes. The effect of other attributes than @name may depend on the file format (see description of file formats below).

If the @name attribute is specified in the message, the newly created tracks are prefixed with the given name. If multiple tracks are created, they are named with the given name as prefix, followed by additional suffixes distinguishing the different tracks depending on the file format (see description of import file formats below). Otherwise, when reading a single track, the given name is simply used as track name. Existing tracks having the same names as newly created tracks are replaced.

The MuBu container supports the following import file formats:

- SDIF, sound description interchange format
- MIDI standard files
- MusicXML
- Plain text files

The data imported from the file is converted into the storage format of the MuBu container. Data that can not be represented by the container is ignored.

SDIF Files

The Mubu container provides a particular support for SDIF files (using <http://sdif.sourceforge.net/>). SDIF files imported by the container are supposed to be exported with a minimum loss of information. Nevertheless, not all SDIF formats are fully supported. Numeric data is stored as 32-bit floating-point numbers and not all textual data can be represented.

When reading an SDIF file, a separate track is created for each combination of an SDIF *stream id*, *frame signature*, and *matrix signature*. If the @name attribute is specified, the names of multiple created tracks are prefixed with the given base name followed by the SDIF stream id and signatures:

< *SDIF stream id* > '-' < *frame signature* > '-' < *matrix signature* >

Otherwise, unnamed tracks are appended to the container.

MIDI Files

Reading a MIDI file, multiple tracks may be created for each track of the MIDI file.

For each MIDI track the following tracks may be created in the MuBu container:

- MIDI notes (all MIDI channels in one track)
- textual meta events (text events, copyright notices, lyrics, markers, and cue points in one track)
- poly pressure events in separate tracks for each MIDI channel and note number
- control change events in separate tracks for each MIDI channel and controller number
- program change events in separate tracks for each MIDI channel
- channel pressure events in separate tracks for each MIDI channel
- pitch bend events in separate tracks for each MIDI channel

If the @name attribute is specified, the names of the created tracks are prefixed with the given base name followed by track identifiers separated by '-'. For files with multiple MIDI tracks (format 1), the first identifier is a MIDI track id (index, *sequence name* or *instrument name*). Only note tracks do not have any additional suffixes.

The other tracks are distinguished with the following suffixes:

- textual meta events: '-text'
- poly pressure events: '-' <channel> '-' <note number> '-pp'
- control change events: '-' <channel> '-' <note number> '-cc'
- program change events: '-' <channel> '-pc'
- channel pressure events: '-' <channel> '-cp'
- pitch bend events: '-' <channel> '-pb'

MusicXML Files

As an experimental feature, a MuBu Container can import *MusicXML* files (using <http://libmusicxml.sourceforge.net/>).

Plain Text Files

Importing a plain text file creates a single track. Each line of the file represents an element of the track. The configuration of the created track can be specified with the message attributes described for the `addtrack` message. For time-tagged tracks, the first column of the file is considered as time-tags. A single column of strings (also mixed with other numeric columns) can be imported as labels when the track is created with the option `@extradata labels`.

refer	name [symbol]	Refers to the MuBu container of the given name. If a mubu object is the only reference to its current container, the current container is saved as "garbage" before the new reference is set. MuBu containers that have been saved as garbage can be restored via a <i>refer</i> message. A warning message is posted into the Max window each time a container is collected as or restored from garbage.
removetrack	track id [atom]	Removes the track of the given id (index or name).
replacetrack	track id [list] max size [int] matrix cols [int] matrix rows [int] attributes [list]	Replaces the given track with an empty track. The arguments and attributes for creating the new track (following the track id), are identical to those of the <code>addtrack</code> message.
track	track id [atom] message [symbol] arguments [list]	Applies a message to the track of the given id (index or name). If 0 is given instead of a valid track id, the message is applied to all tracks of the container. Apart from name, all track messages are also messages of mubu.track .

Valid track messages are (see [mubu.track](#) for the description of message arguments):

- name — sets track name or remove track name (without argument)
- set — sets track element(s) at given index
- insert — inserts track element(s) at given index

append — appends element(s) to track
remove — removes element(s) from track
insertpoints — inserts time-tagged element(s) into track
appendpoints — appends time-tagged element(s) to track
removepoints — removes time-tagged element(s) from track
clear — clears track (removing all elements)
getinfo — outputs track info (all or for a given key)
info — sets track info from a list of key and value pairs
size — sets track size
matrixrows — set number of matrix rows at given index (only if **matrixrowsvar** is enabled)
matrixcolnames — sets matrix column names
samplerate — sets sample rate
sampleperiod — sets sample period
sampleoffset — sets sample offset
silent — enables or disables updates to track listeners (e.g. [imubu](#))
getsize — outputs track size
getmatrixcols — outputs number of matrix columns
getmatrixrows — outputs number of matrix rows
getmatrixcolnames — outputs matrix column names
getmatrixrowsvar — outputs whether track allows track elements for having a variable number of matrix rows
getextradata — outputs track extra data type
gettmetagged — outputs whether track has time-tagged data
getsamplerate — outputs sample rate
getsampleperiod — outputs sample period
getsampleoffset — outputs sample offset
getsilent — outputs whether track updates of track listeners are enabled

write	file path [symbol]	Exports container data to a file (not yet implemented).
--------------	--------------------	---

Attributes

<i>Name</i>	<i>Type</i>	<i>g/s</i>	<i>Description</i>
bufferindex	int		Changes the current buffer. Most of the messages sent to a MuBu container, concern only the current buffer.
embed	int		Enables/disables the conservation of the container data in copy/paste and when saving the patcher. Attention: This option should only be used for containers with a relatively small amount of data. Larger amounts of data should be saved to a file (see write).
name	symbol		Sets the container name. If a container of the given name exists the mubu object refers to that container (same as the message refer). Otherwise, and if the mubu object does not share its container with other mubu or imubu objects, the current container is simply renamed. If the mubu object shares its current container with other mubu or imubu objects and the given

name does not refer to an existing MuBu container, a new empty container is created with the given name.

numbuffers	int	Sets the number of buffers of the container. If new buffers are created, the additional tracks are created with the same maximum size (capacity) as the tracks of the last buffer.
------------	-----	---

[Information for box attributes common to all objects](#)

Examples

See Also

<i>Name</i>	<i>Description</i>
imubu	Graphical interface around a MuBu container
mubu.track	MuBu track reference