AtlasHelp

AtlasHelp

Publication date 2012

Table of Contents

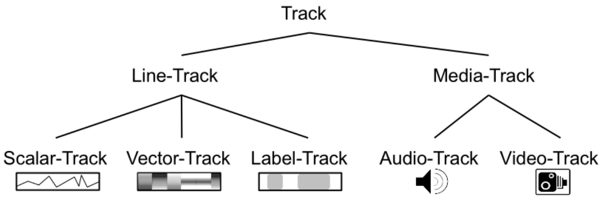
List of Tables

1 Introduction

Atlas was developed to allow labeling of multimodal data from experiments (i.e. video and audio), but also other physical input types i.e. heart rate information.

The main aim of Atlas is to provide a graphical representation of input data and to allow the user to label provided data with attributes. Additionally Atlas allows auto-labeling by using existing labels and searching for matches in the feature input. Features used for such auto-labeling can either be raw input data from the experiment or data derived from such raw data.

All input data in Atlas is considered a track. The structure of track types is shown in the image below:



Tracks in Atlas

Tracks can be divided into media-tracks (video and audio) and line-tracks (also called data-tracks). Video-tracks are displayed using the VLC media player. Audio-tracks also use VLC and should be supplied in WAV format which enables Atlas to create a scalar-track as visual representation of the audio input (without the audio's scalar-track Atlas can only use the audio-track for playback).

Line-tracks are shown in Atlas' LabelArea, an area for visual represention of input data. Scalar-tracks are two-dimensional (value at a specific time) input formats like audio amplitude. Vector-tracks are multidimensional graphical formats (vector at a specific time). Vector-tracks are often used to import more complex data that is created from the experiment's original output.

Label-tracks are user defined tracks to mark wanted properties in the input tracks. For example a label-track might show a test subjects emotional state. The user can create so called label-classes with respective entities in order to label certain monitored characteristics. For example after creating a label-class 'emotional state', the user can create classes' entities (i.e. 'happy' or 'sad') and label the input data in respect to the state of the test subject.

In auto-labeling mode the user assigns a feature-track to a label-class. Atlas will then attempt to automatically create labels for the label-track with the supplied feature-track as input reference.

The combination of all input types and LabelTracks is called 'collection'.

Table 1.1. Track overview

Track	Use
Video	Video tracks are played back using the VLC media player. VLC supports playback of many video codec formats.
Audio	Audio-track playback is supported for WAV and mp3 formats. When using WAV as input, Atlas can generate a scalar-track of the input file. Wav input is therefore preferable. Please note

Track	Use
	that it is possible to transcode other audio formats to wav with many audio tools.
Scalar	Two dimensional input (value at a specific time). Scalar-tracks are optionally created when importing wav files.
Vector	Multidimensional input (vector at a specific time). Used for more complex input types.
Label	User defined track to allow labeling of criteria the user is interested in. Each Label-track uses a label-class representing a characteristic that is monitored. In the label-track labels with entities (different predefined states of the monitored characteristic) are used.

2 Quickstart

This chapter shows basic use of Atlas by demonstrating how to start the program and how to exemplarily work with a single video as collection and how to add labels with the label-class generic. The idea is to provide a simple scenario to allow insight into the workflow of Atlas. Further options will be explained in the following chapters.

2.1. System Requirements

Atlas needs a Java Virtual Machine version 1.6 or newer. Additionally to support video and audio playback, Atlas uses a native installation of the VLC media player. There is a legacy version of Atlas that supports VLC version 1.x with certain operating systems. It is nevertheless highly recommended to use VLC version 2.x with the respective Atlas version, because it is more stable. It also supports more operating systems. The legacy version will not be further supported and be dropped at some point in the future.

Chapter Known Issues provides help if Atlas does not work on your system. It also contains a section about (Un)Successfully tested Operating System, Java and VLC Versions .

Depending on the input types of projects, a not completely outdated personal computer may be enough to use Atlas fluently. When working with greater amounts of input data like several video files from different angles, state of the art hardware with current multi-core processors and more than 4 GB of memory may be required to allow work in an acceptable pace. When using several video inputs, support of hardware video-decoding can considerably reduce strain on the CPU while fast solid state disks provide the data transfer rates needed for smooth playback.

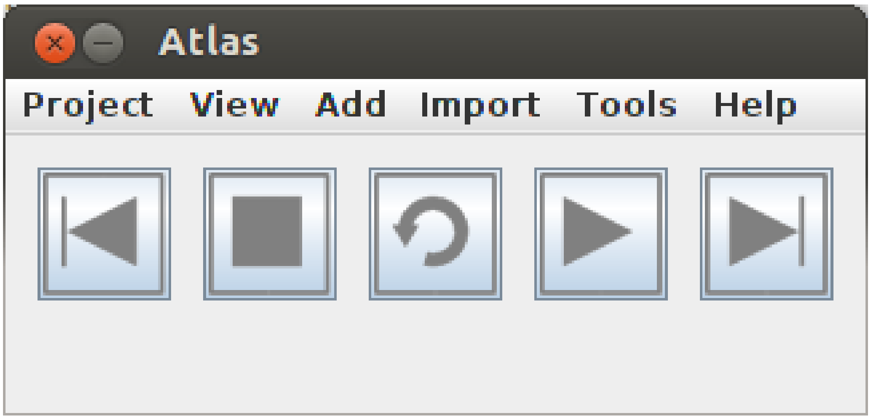
2.2. Starting Atlas

If .jar files are associated with a Java Runtime, starting Atlas should be possible by double clicking the "Atlas.jar" file. From Windows Command Prompt or a Linux shell the program can be started by the command

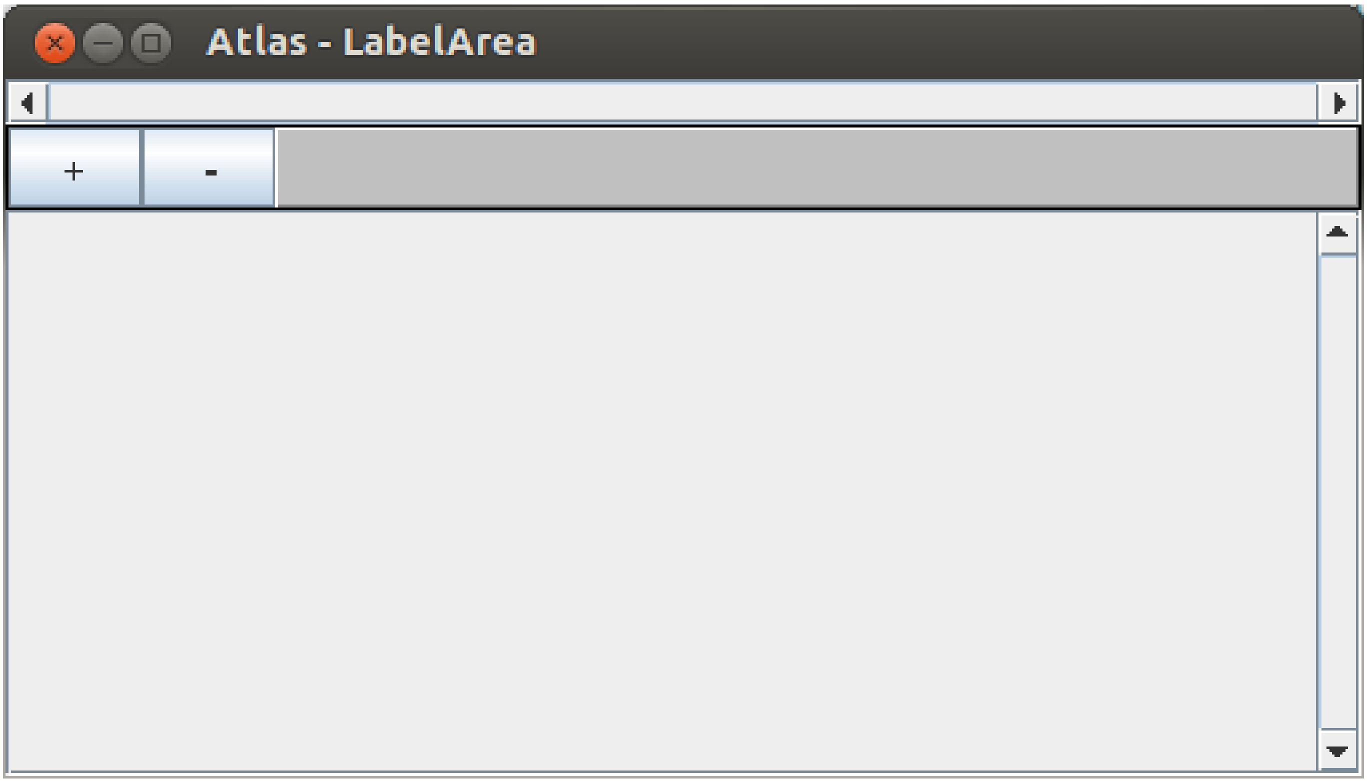
java -jar <PATH TO Atlas.jar> (i.e. java -jar c:\Users\exampleuser\Desktop\Atlas\Atlas.jar)

An initial warning that Atlas is not able to read 'Atlas.properties' can be ignored.

Two windows will be displayed: 'Atlas' and 'Atlas - LabelArea'.



Atlas main window



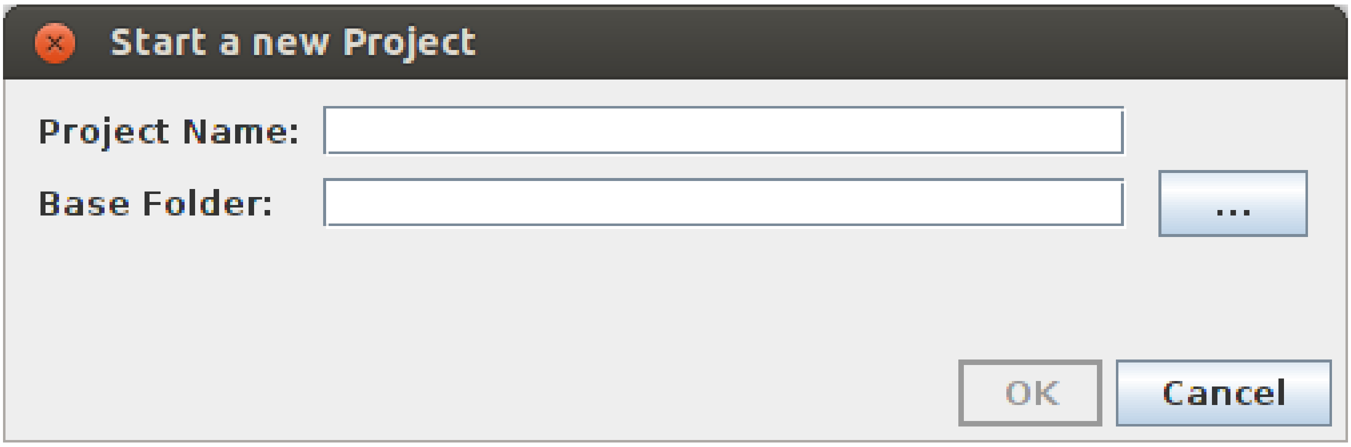
Atlas - LabelArea

2.3. Creating a new Project

Atlas organizes data belonging together in projects. To start a new project select

Atlas → Project → New,

which will start a dialog requesting input of a projectname and a base folder for the project. Atlas will create a subfolder in the base folder with the projects name for all relevant project data. After selecting 'OK' Atlas creates the subfolder in Base Folder.



Start a new project dialog

The XML file with the project's name in this subfolder can be used to load the project after Atlas restarts.

Atlas → Project → Load

Saving a project manually can be done with

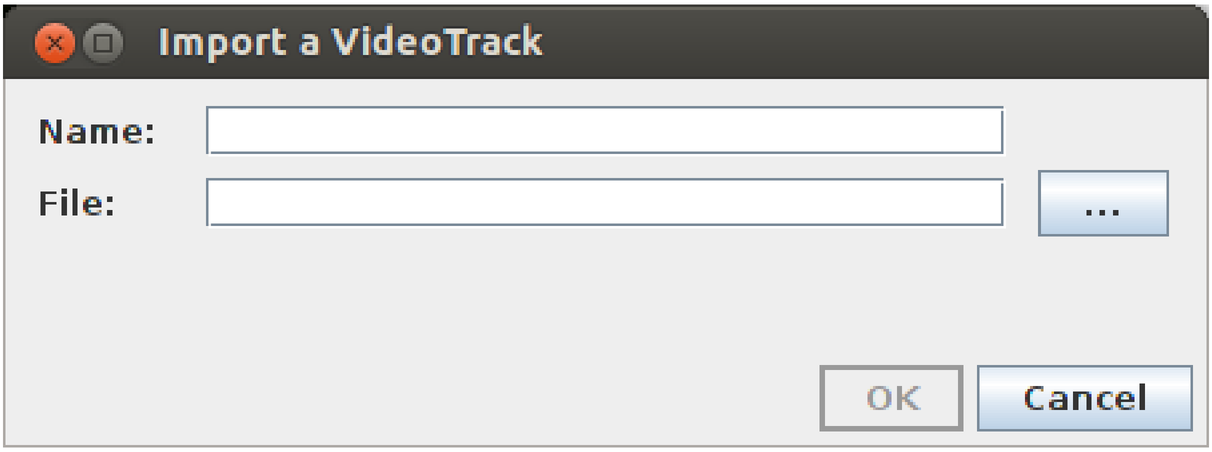
Atlas → Project → Save

Atlas will always ask if the project should be saved before exit or switch to another project.

2.4. Adding a video source

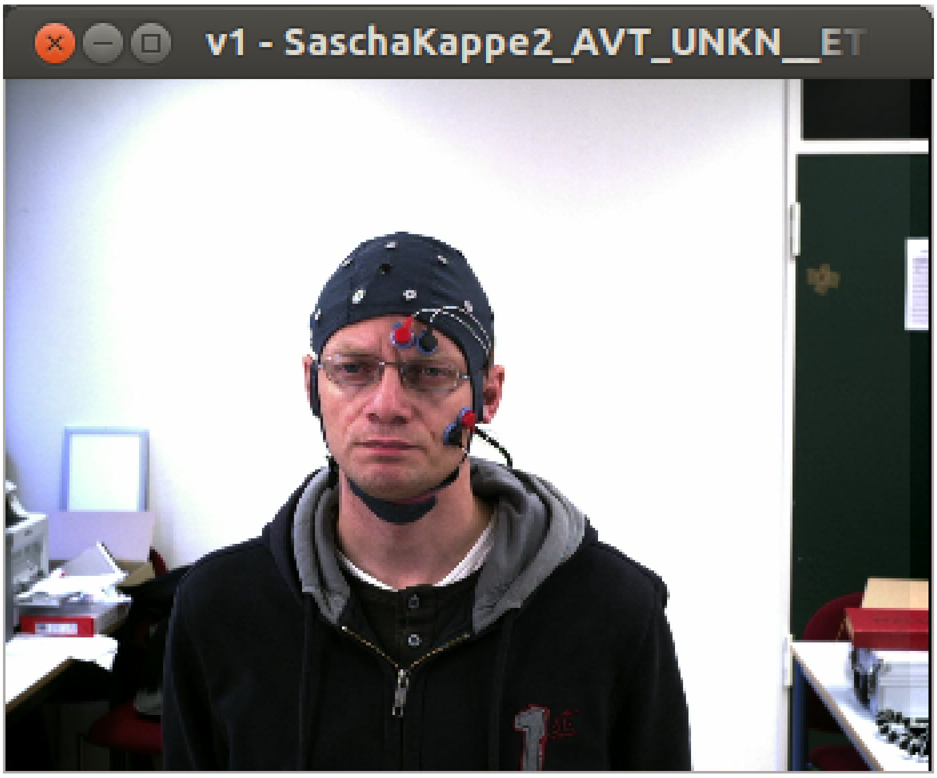
At this point a video source can be added through

Atlas → Import → Video.



Import a VideoTrack

To import a video a name and the video's path must be supplied. After confirming with 'ok', a new window with the video will be displayed.



An Imported VideoTrack

Note

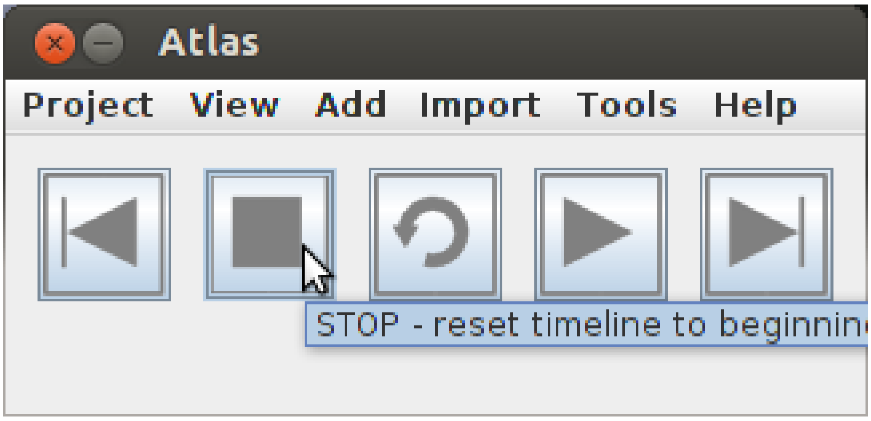
Should Atlas crash when importing a video, the most likely problem is either a corrupted video file or an incompatibility between the VLC media player, the Java Virtual Machine and the operating system. See chapter

'Known Issues' section 'Video Imports General Guidelines' for possible workarounds AFTER checking with other video sources. A Table with tested systems is provided in 'Known Issues' section ' (Un)Successfully tested Operating System, Java and VLC Versions' .

Warning

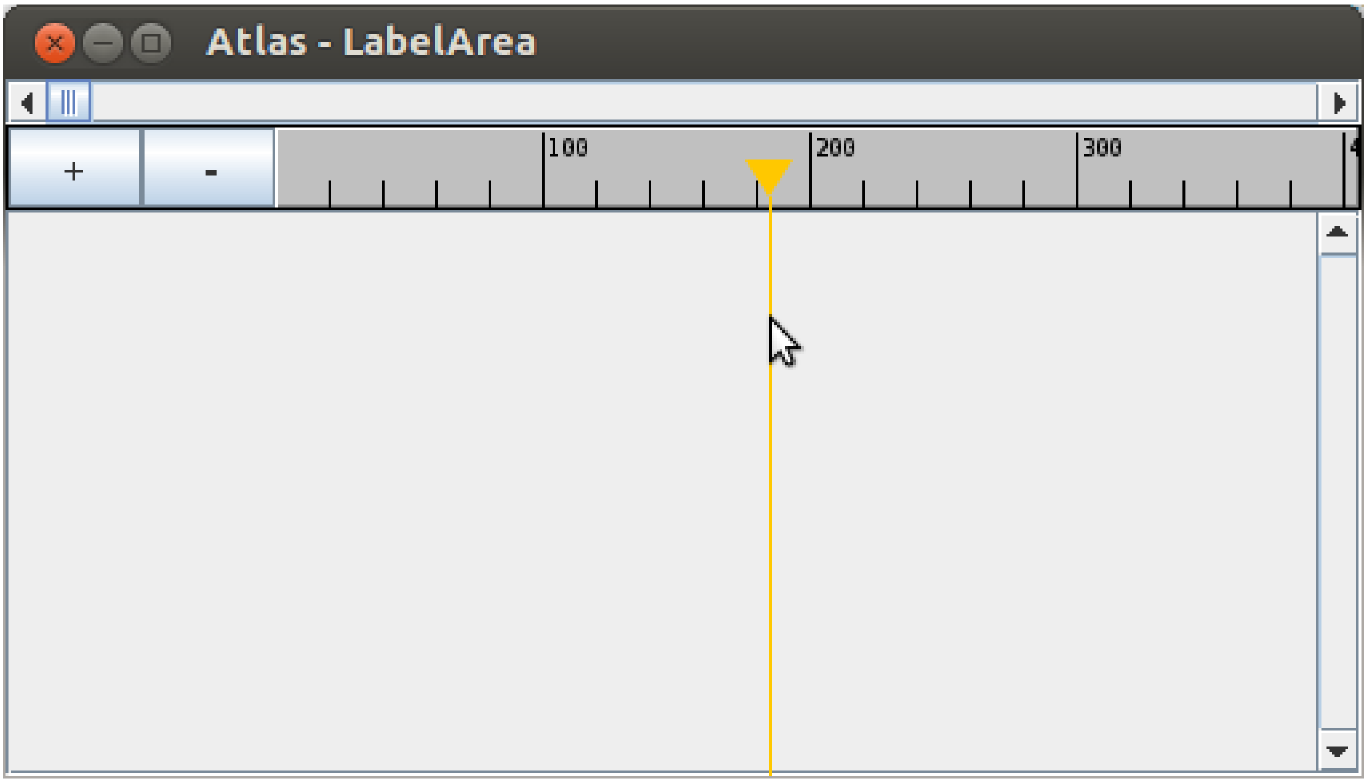
Atlas can only be closed from main window or LabelArea. Closing a video window by clicking the close button will not close the window, but the VLC media player will be ended - this is an issue that may never be fixed. See chapter 'Atlas Main Window' section 'View' → 'TrackEditor' on how to remove a (video) track or disable its playback.

It's possible to use all media control buttons but 'Loop selection', which will play the collection with start and end of a selected label in a loop. The buttons purposes are displayed when holding the mouse arrow over them.



Media control buttons - mouse over will show button effects

The projects timeline can be adjusted with the mouse by selecting the desired point in time in the LabelArea:



Adjusting timeline in LabelArea

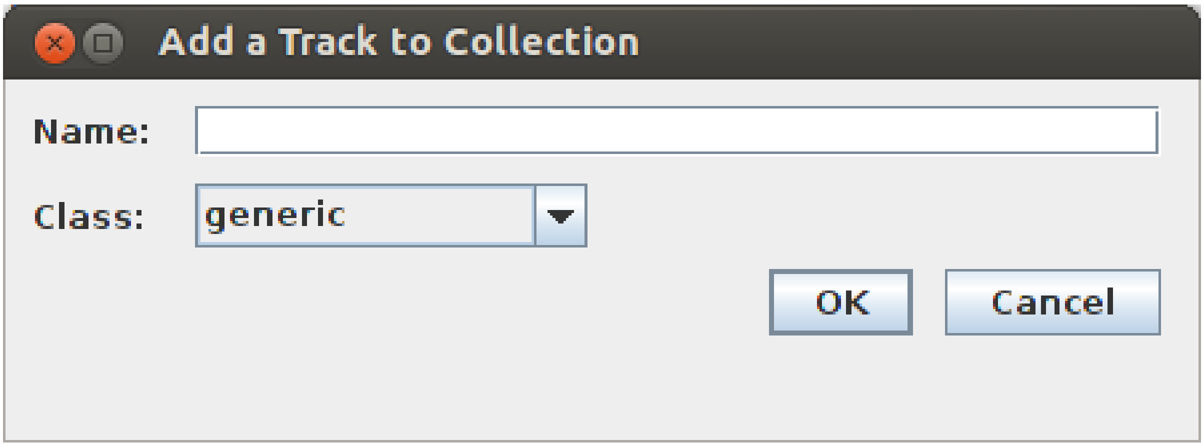
The scaling of the timeline can be adjusted with the mouse wheel as well as the '+' and '-' buttons in the top left corner of the LabelArea.

2.5. Adding a LabelTrack

In order to label sections of the video, a LabelTrack can now be added through

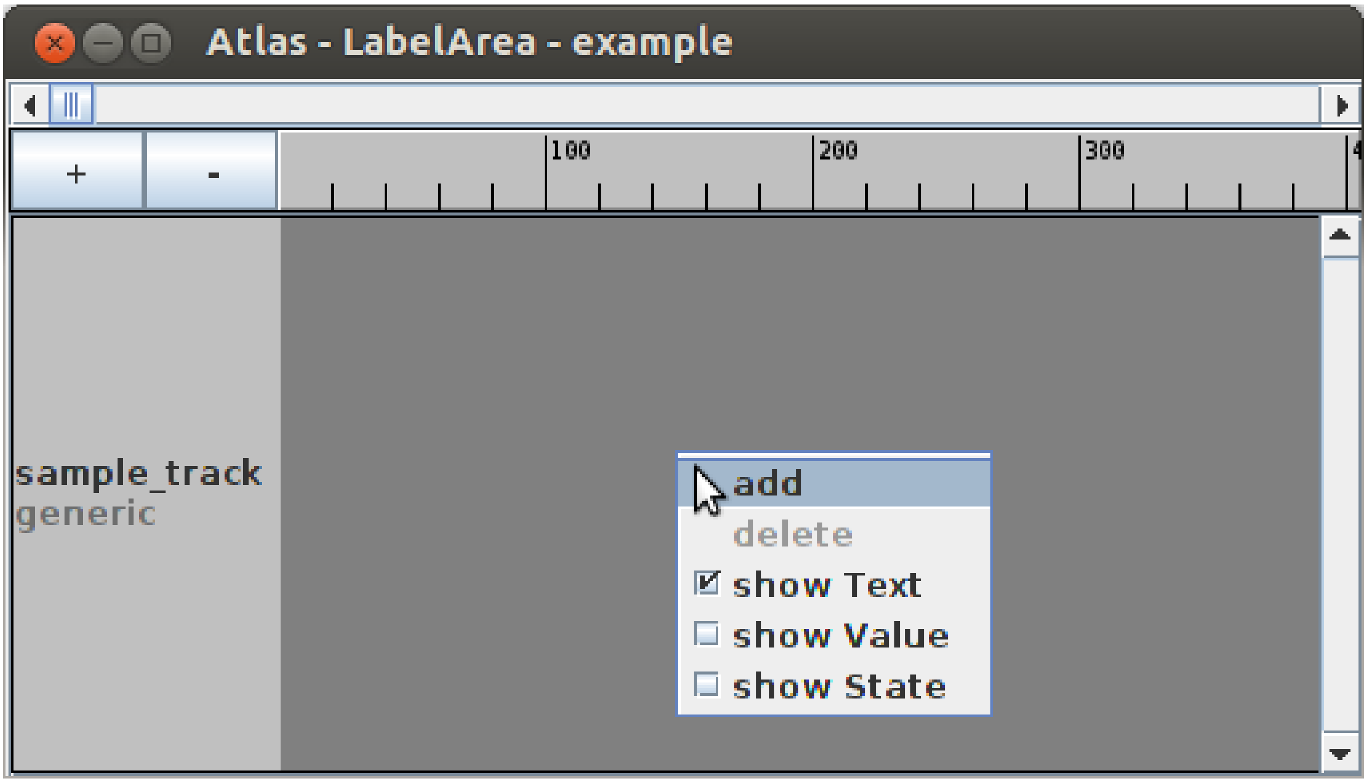
Atlas → Add → empty LabelTrack.

A name must be assigned to the track for identification. The label-class 'generic' that is available by default and offers the entity 'labeled' will be used for now.



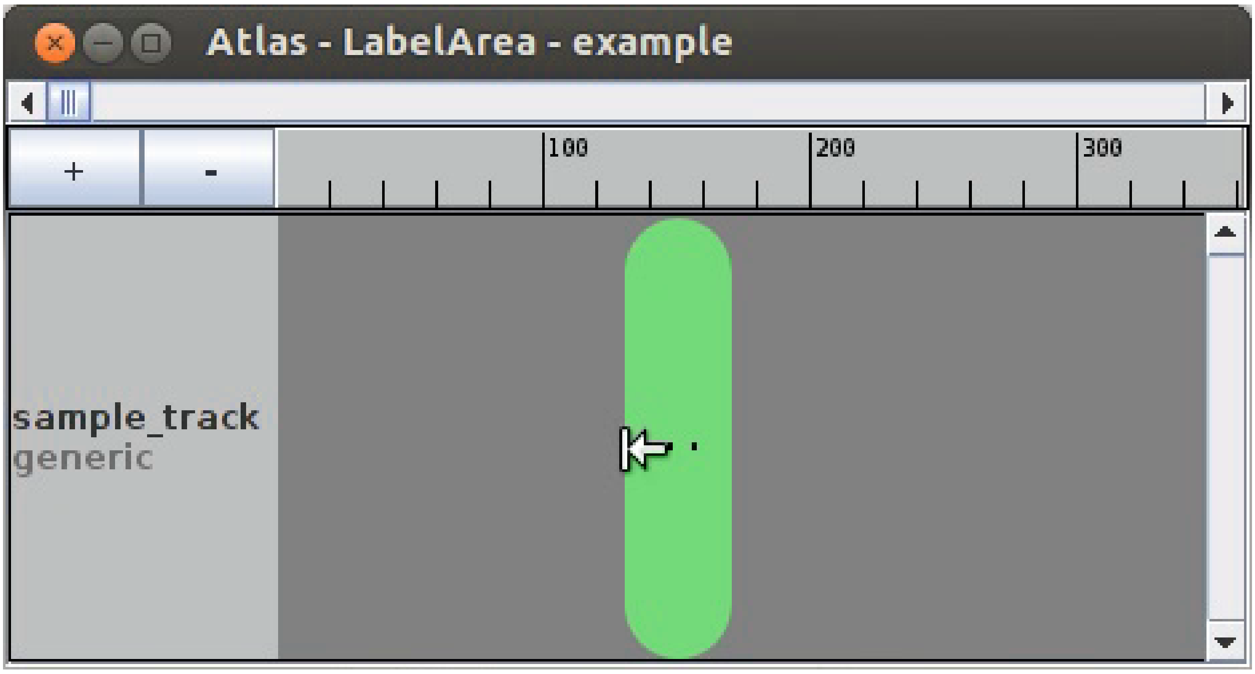
Adding a LabelTrack to the Collection

The LabelArea now contains a field for the label-track. A label can be added with the mouse by right-clicking at the desired section and selecting 'add'.



Adding a label to a LabelTrack

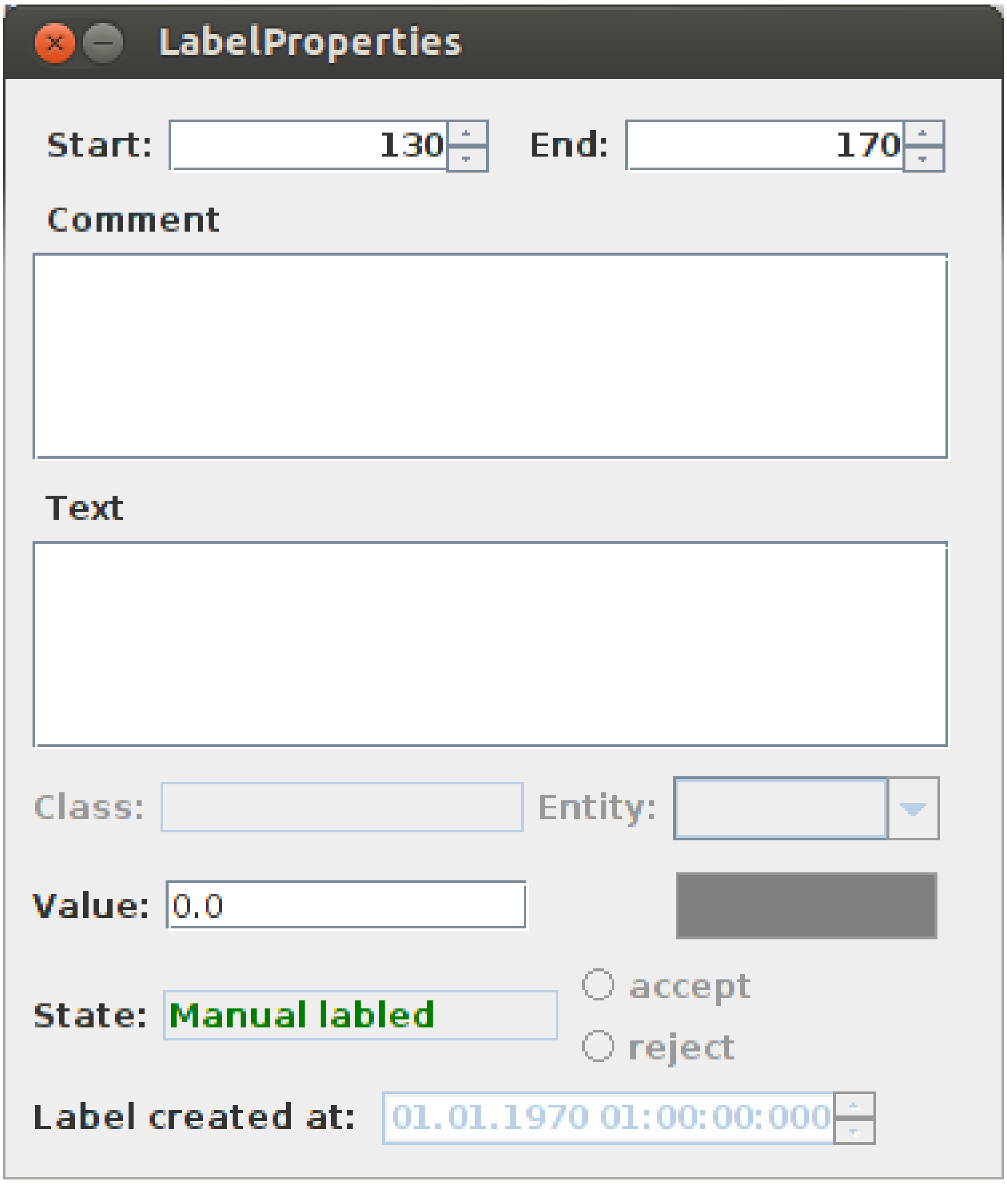
Now the labels start and end can be adjusted to the wanted position by Drag & Drop. Remember that you can use the mouse wheel or '+' and '-' buttons to achieve the desired scaling. It is also possible to use 'Loop-Playback' when the label is selected. Loop-Playback will continuously playback the tracks using start and end time index of the selected label.



Adjustment of Label start and end using Drag & Drop

The label can be removed using right-click → delete.

The labels can be edited using Atlas → View → LabelProperties.



LabelProperties

Start: 130 End: 170

Comment

Text

Class: Entity:

Value: 0.0

State: Manual labled ○ accept ○ reject

Label created at: 01.01.1970 01:00:00:000

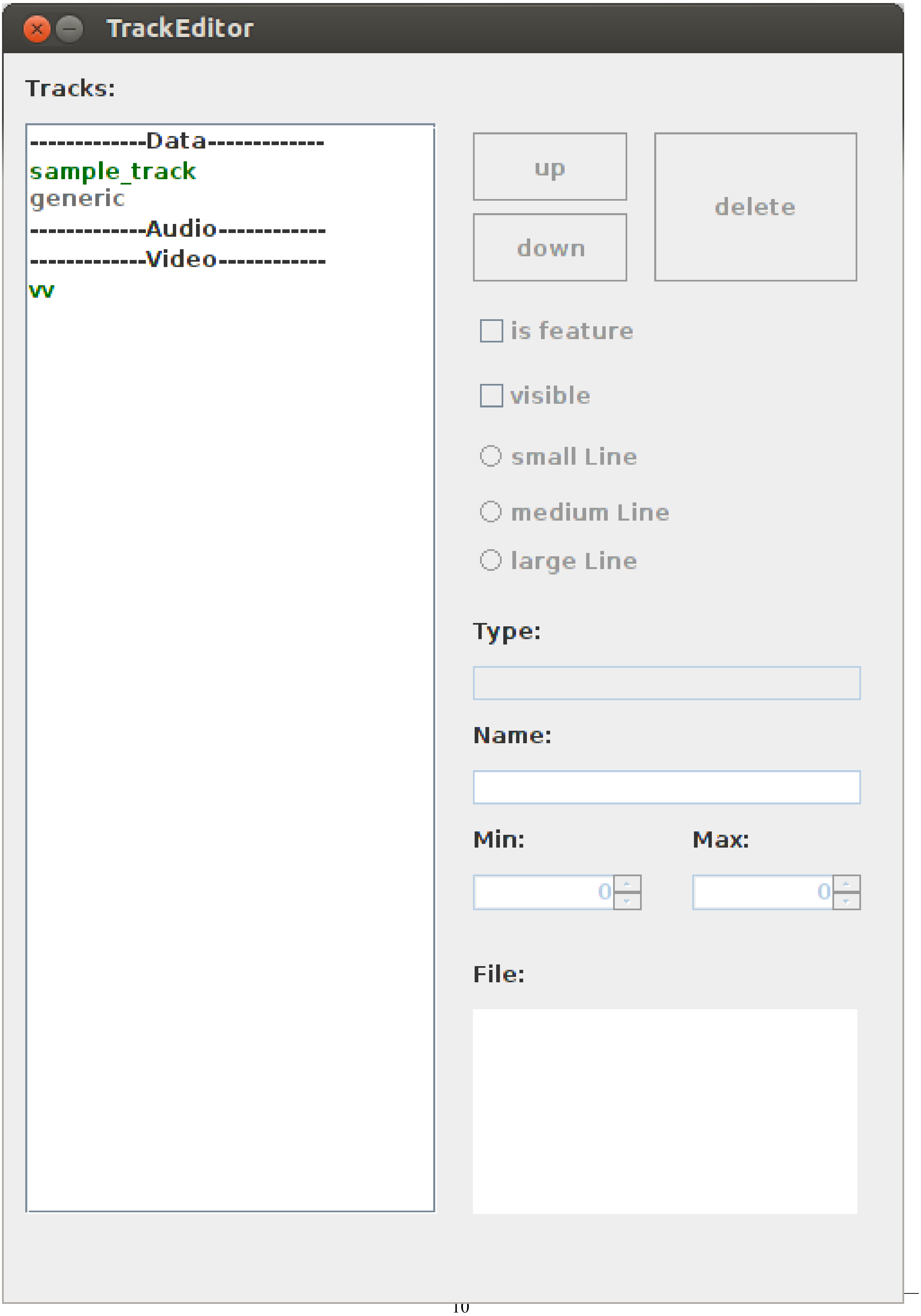
LabelProperties of the created label

The LabelProperties window shows the properties of the selected label in the LabelArea. LabelProperties are more thoroughly described in chapter 'Atlas Main Window' section 'View' → 'LabelProperties'.

2.6. TrackEditor

The current project can now be viewed in the Trackeditor:

Atlas → View → TrackEditor



-------------Data-------------
sample_track
generic
-------------Audio------------
-------------Video------------
vv

Min: 0

Max: 0

TrackEditor of example project

The TrackEditor is used to manage all tracks of the project - in this case the video track and the label track. Working with the TrackEditor is more thoroughly described in chapter 'Atlas Main Window' section 'View' → 'TrackEditor'.

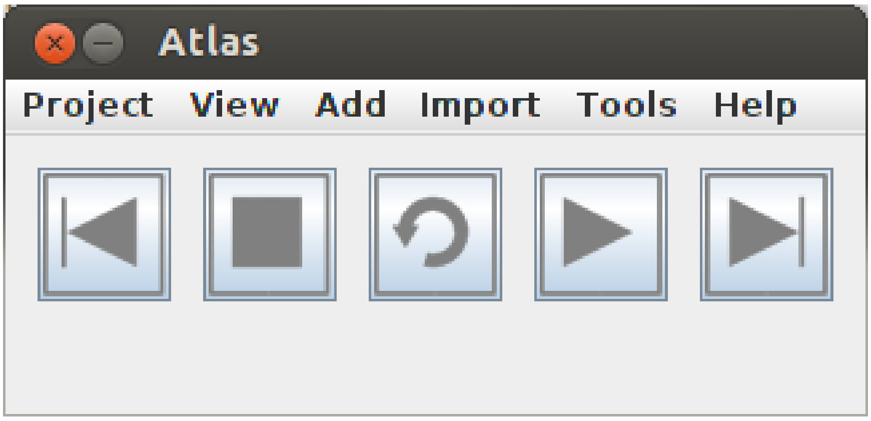
2.7. Summary

At this point we covered creating a collection with with a single video input file, a label-track with label-class generic, a manually created label and the project's track overview in the TrackEditor. Now additional input files or label-tracks could be added. The quickstart chapter hopefully gave an introduction on how to add file and label-tracks as well as how they are organized in an collection.

A more in-depth view for the Atlas' features is provided in the following chapters.

3 Atlas Main Window

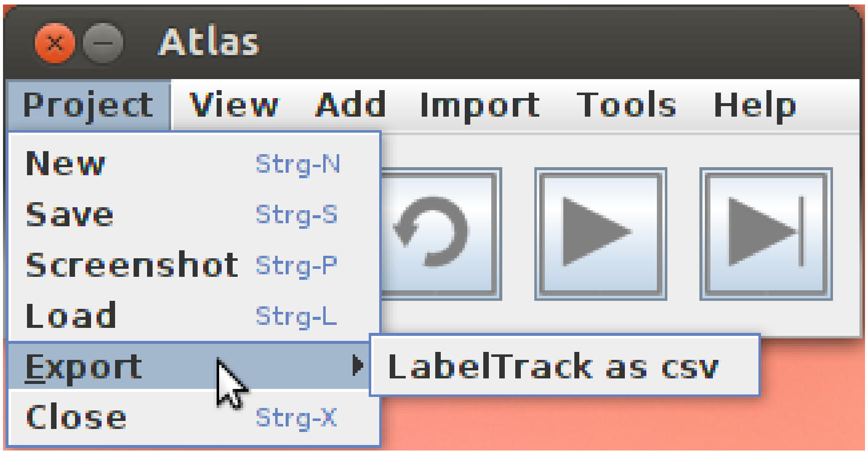
The Atlas main window offers possibilities to work with collections. It concerns the management of projects, tracks, label-classes and as user interface options as well as tools useable in Atlas i.e. for auto-labeling.



The Atlas main window

3.1. Project Menu

The Project menu basic options like creating, saving and loading projects as well as exporting LabelTracks in csv format.

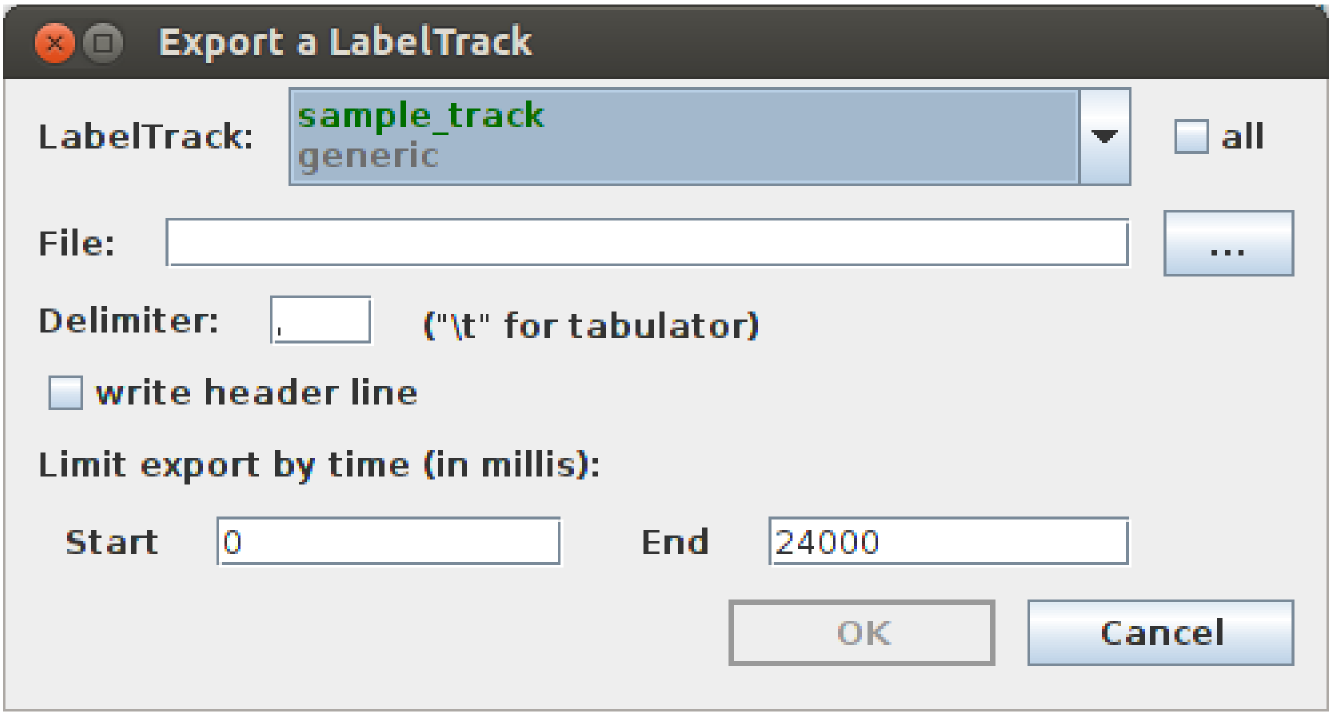


Project Menu

Table 3.1. Project Menu

Item	**Use**
New	Create a new project. A name and a base directory are needed. Atlas will create a subfolder in the base directory that will be used for managing the project.
Save	The project's currents state is saved.
Screenshot	A screenshot of the LabelArea will be created.
Load	Loads a saved project.
Export	Exports from project. Currently only an option for csv export of LabelTracks available.
Close	Closes Atlas.

The Export option allows to export a LabelTrack in csv format:



Export a LabelTrack

LabelTrack: sample_track generic

□ all

File:

Delimiter: ,

("\t" for tabulator)

□ write header line

Limit export by time (in millis):

Start 0

End 24000

Export of LabelTrack as csv

The dialog allows the user to select

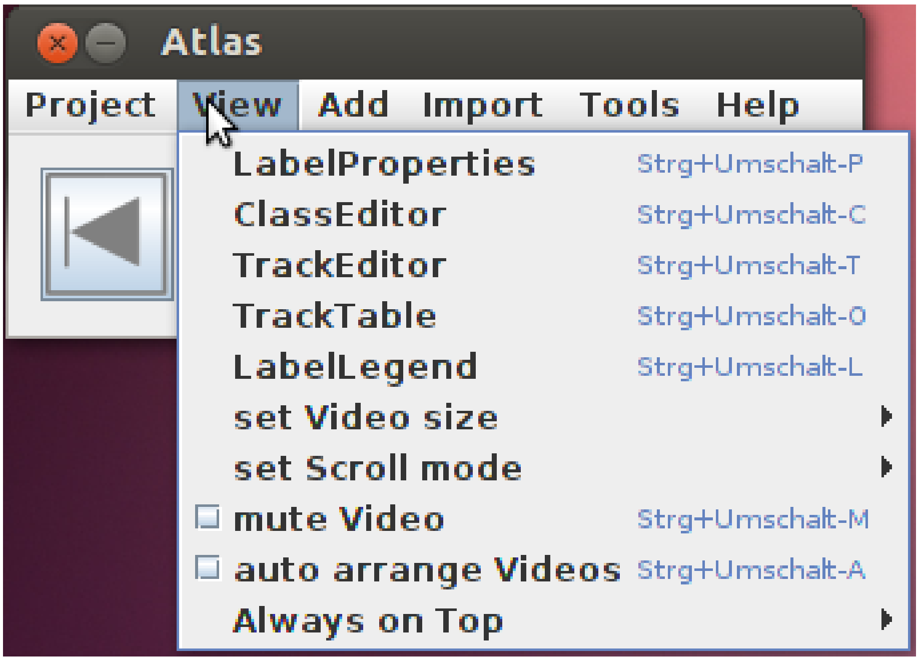
- a specific or all LabelTracks,
- name of the csv file,
- the delimiter used in the file,
- the inclusion of a header line and
- the selection of labels only between selected time values.

This way LabelTrack export for external applications is possible.

3.2. View Menu

The windows LabelProperties to LabelLegend concern status and options of labels, label-classes, and tracks in the collection. They allow manipulation of such values.

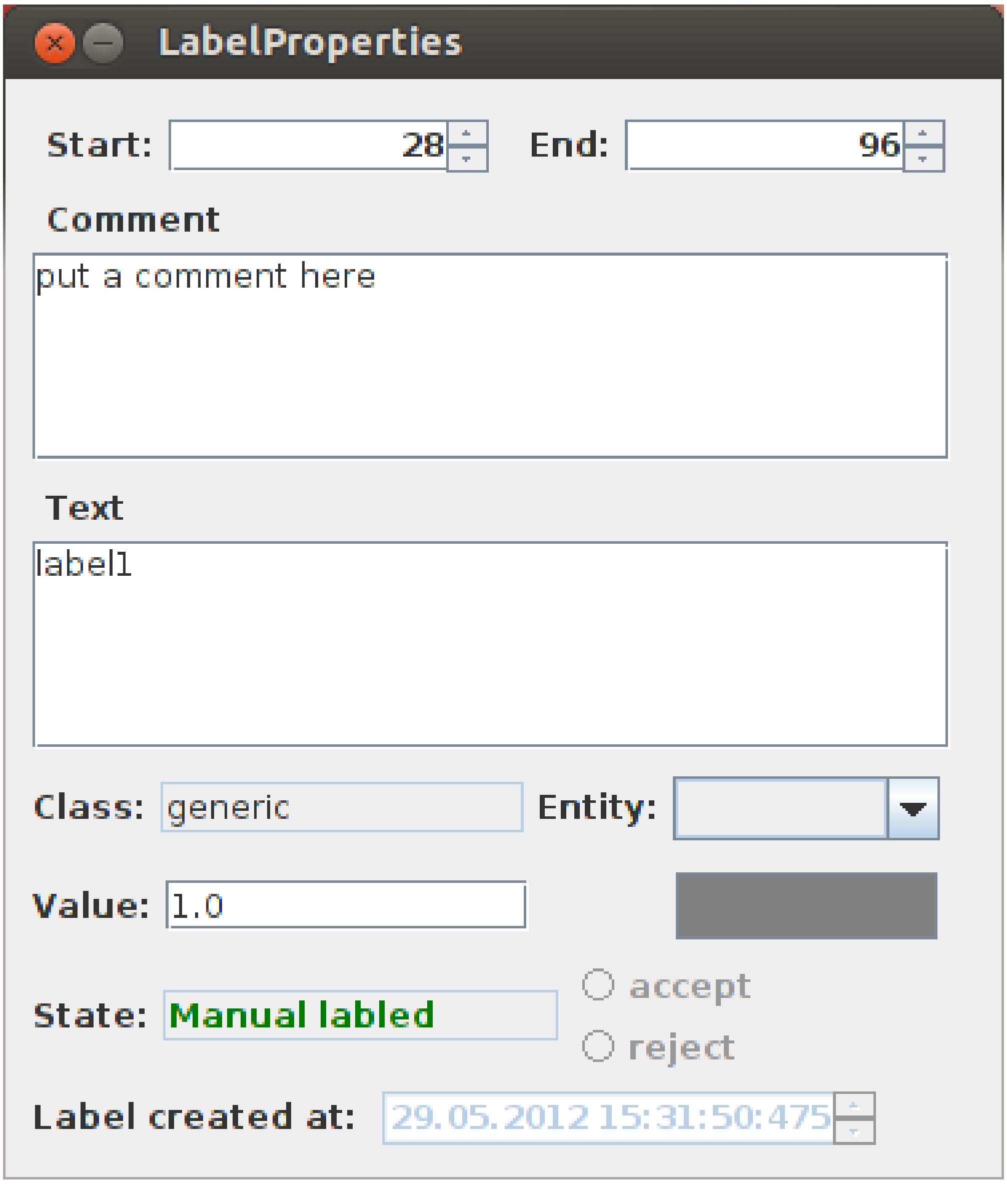
The remaining options are convenience options for the user like changing the size of displayed videos.


Atlas
Project
View
Add
Import
Tools
Help
LabelProperties Strg+Umschalt-P
ClassEditor Strg+Umschalt-C
TrackEditor Strg+Umschalt-T
TrackTable Strg+Umschalt-O
LabelLegend Strg+Umschalt-L
set Video size
set Scroll mode
□ mute Video Strg+Umschalt-M
□ auto arrange Videos Strg+Umschalt-A
Always on Top

View Menu

3.2.1. LabelProperties

The LabelProperties window is used to view and set information concerning the currently selected label in the LabelArea.



Start: 28 End: 96

Comment

put a comment here

Text

label1

Class: generic Entity:

Value: 1.0

State: Manual labled ○ accept ○ reject

Label created at: 29.05.2012 15:31:50:475

Label Properties

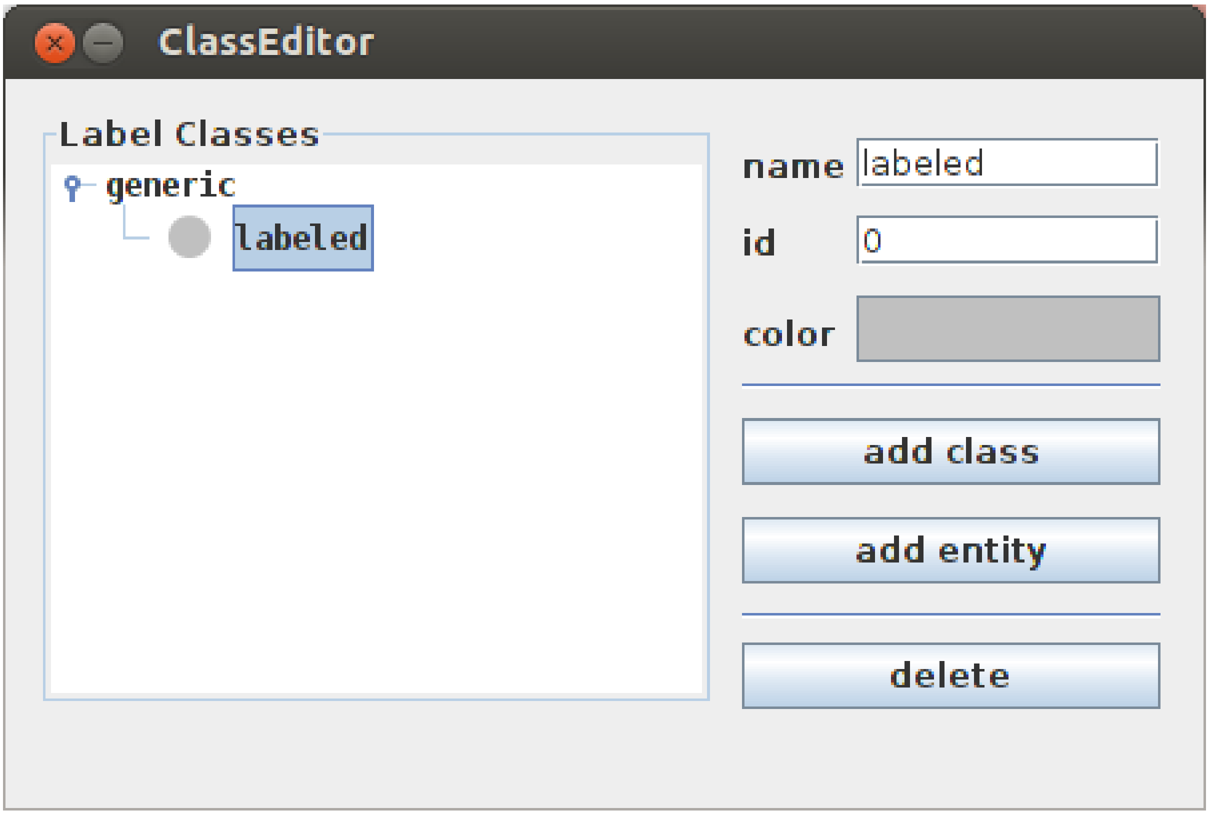
Table 3.2. Label Properties

Component	Use
Start / End	Set start and end time-index of label.
Comment	Textfield to add a comment.
Text	Main text entry of the label. Will be displayed in the LabelArea.
Class	Shows the class of a label (see ClassEditor for more information on label classes).
Entity	Every LabelClass has a set of entities that allow to set a specific characteristic the user wants to monitor. The only entity in generic class is 'labeled'. Entities are defined in the ClassEditor (see ClassEditor).
Value	A numerical value to describe the label if applicable, i.e. to express the intensitiy of an entity.
Color field	Shows the entitie's color.

Component	Use
State	Indicates if the label was created by the user or proposed by Atlas (see Suggest Labels). The states are 'manually labeled' and 'auto-labeled' respectively and cannot be changed.
accept / reject	If a label is 'auto-labeled', a label can be accepted as correctly labeled or rejected. Must be done manually for all 'auto-labeled' labels.
Label created at	Timestamp of label creation.

3.2.2. ClassEditor

The ClassExitor is used to define classes for labeling. A class has a name and entities that represent the states of labels.



name labeled

id 0

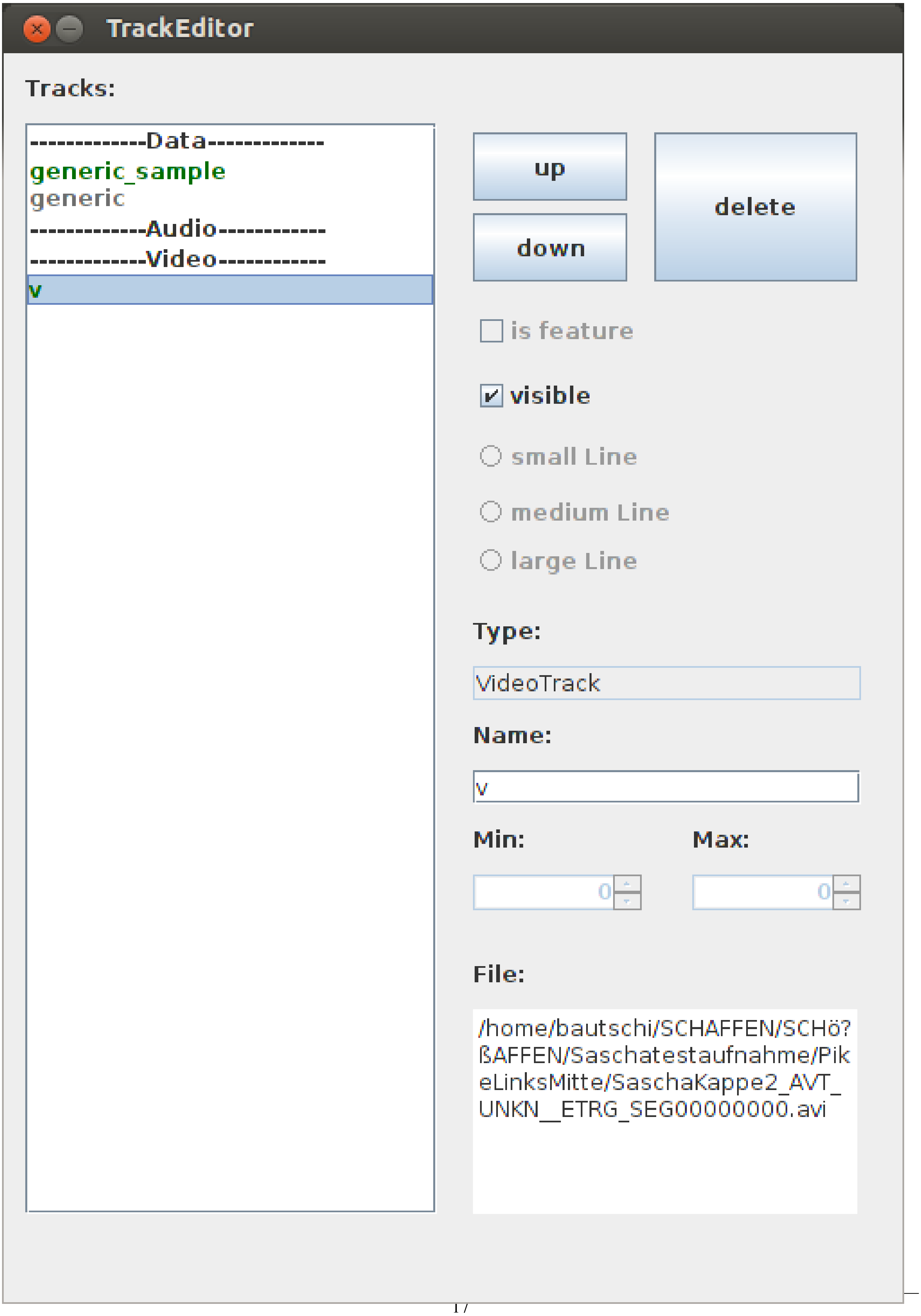
ClassEditor

The ClassEditor is used to add and delete label-classes and their entities. The window allows to adjust name, id and color of existing entities. Renaming a LabelClass is not possible, it has to be deleted and added with the new name (and entities).

The LabelClasses are then used when adding LabelTracks to allow adding labels and setting the respective entities in the labels.

3.2.3. TrackEditor

The TrackEditor is used to manage existing tracks. It is used to change the order of tracks or delete tracks from a collection when they are not wanted any more. A few properties of tracks, like the name, can be adjusted in the TrackEditor, too.



TrackEditor

Tracks:

-------------Data-------------
generic_sample
generic
-------------Audio------------
-------------Video------------
v

□ is feature

☑ visible

○ small Line

○ medium Line

○ large Line

Type: VideoTrack

Name: v

Min: 0

Max: 0

File: /home/bautschi/SCHAFFEN/SCHö?ßAFFEN/Saschatestaufnahme/PikeLinksMitte/SaschaKappe2_AVT_UNKN__ETRG_SEG00000000.avi

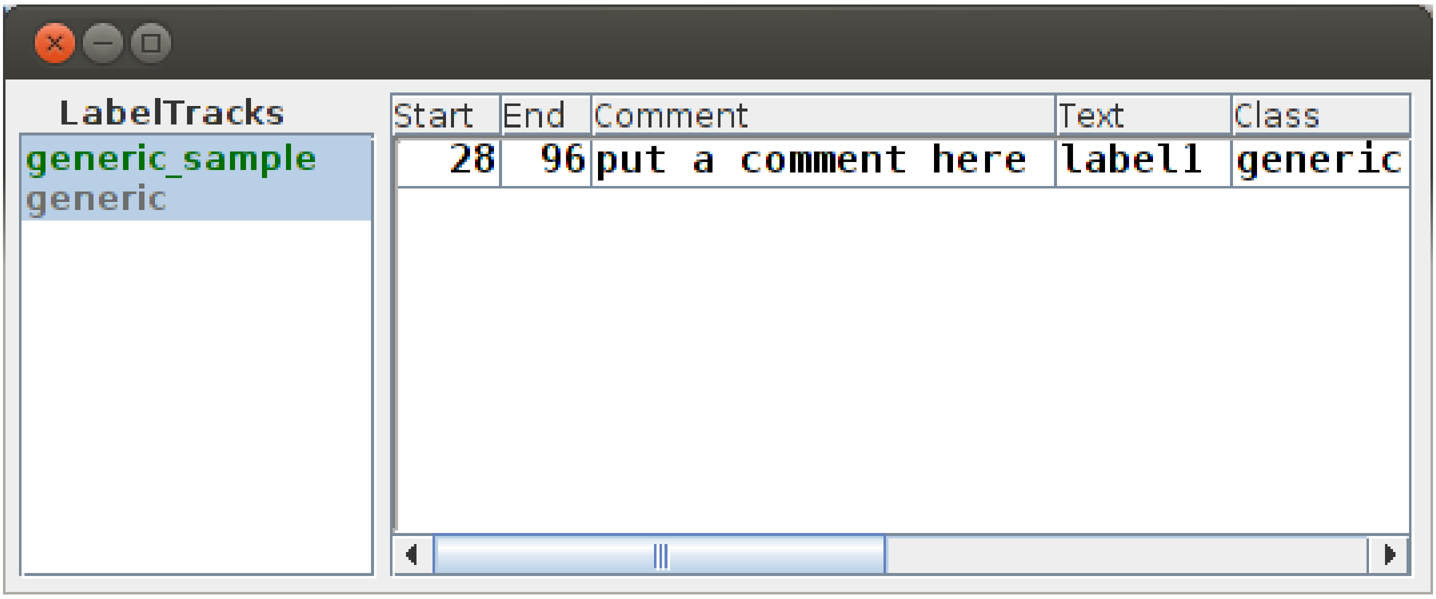
TrackEditor

Table 3.3. TrackEditor

Component	Use
Tracks	A list of all existing tracks in a collection. Tracks are grouped as data-tracks (scalar-, vector- and label-tracks), audio-tracks and video-tracks.
up / down	Used to order tracks of a group (data, audio or video).
delete	Remove the selected track from the collection. Data-tracks keep their files in the project folder.
is feature	Mark this track to allow using it in Atlas' auto-labeling feature Suggest Labels .
visible	Show a track in the collection. Unchecking the box will hide the track from the user.
small/medium/large Line	Allows the change the height of tracks in the LabelArea.
Type	The tracks type (as described in the introduction). Cannot be changed by user.
Name	See / adjust the user-defined name of the track.
Min / Max	Minimal and maximal display value for scalar-tracks and vector-tracks in LabelArea.
File	Shows where the tracks file is located. The created by Atlas are located in the projects folder.

3.2.4. TrackTable

'TrackTables' will display a window with all label-tracks. All labels of a selected label-track are displayed. There are no editing possibilites in this window, but all labels of label-tracks can be viewed.

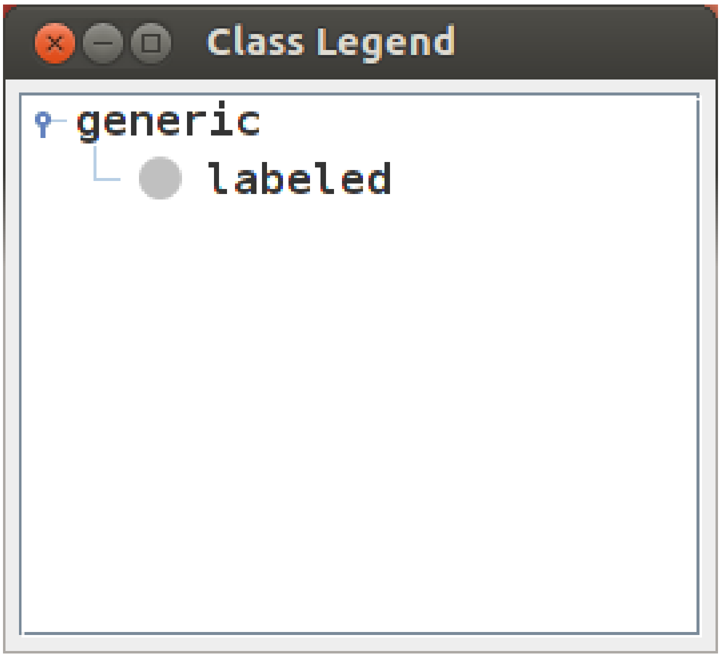


Start	End	Comment	Text	Class
28	96	put a comment here	label1	generic

TrackTable

3.2.5. LabelLegend

The LabelLegend shows an overview of existing label-classes and their entities (see ClassEditor for information on label-classes and entities).



LabelLegend

3.2.6. Convenience Options

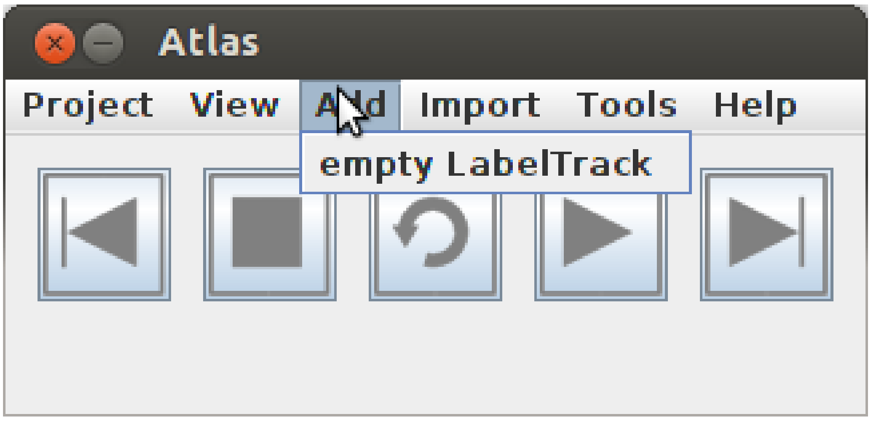
The last four options are used for user convenience.

Table 3.4. Convenience Options

Item	Use
set Video size	Sets the size if video input files. Used to adjust size if too small or too big.
set Scroll mode	Set the way the tracks in the LabelArea are scrolled during playback when the end of the displayed area is reached. When using *half jump* , the end time-index will jump to the middle of the displayed area, while with *full jump* the end time-index will be moved to the start of the displayed area. *Slide* will keep the current time index in the middle of the area while the tracks slide through.
mute Video	Disables audio output of VLC media player.
auto arrange videos	Atlas tries to arrange the videos so they don't overlap.
Always on Top	Allows to keep selected windows in the foreground.

3.3. Add Menu

The Add menu is used to add label-tracks to a collection.



Add Menu

Table 3.5. Add Menu

Item	Use
empty LabelTrack	Add a new empty label-track to the collection.

3.3.1. empty LabelTrack

For adding an empty label-track to the collection a name and a label-class must be provided. Label-classes are managed in the ClassEditor (see ClassEditor). The label-track will then allow to use the classes entities to assign the respective states.

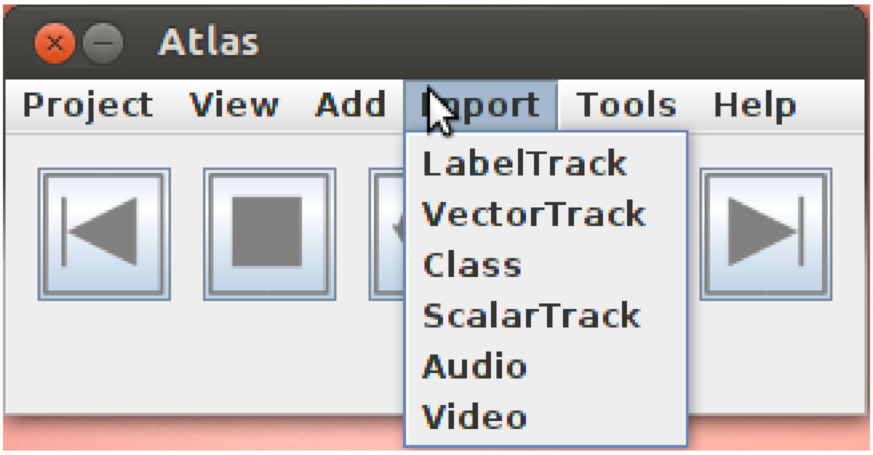
Table 3.6. empty LabelTrack

Item	Use
name	Supply a name for the label-track.
class	Select an existing label-class for the new label-track. Label-classes are managed in the ClassEditor (see ClassEditor).

3.4. Import Menu

Atlas can import existing data. Those imports include

- label-tracks,
- vector-tracks,
- (label-)classes,
- scalar-tracks,
- audio-tracks and
- video-tracks.



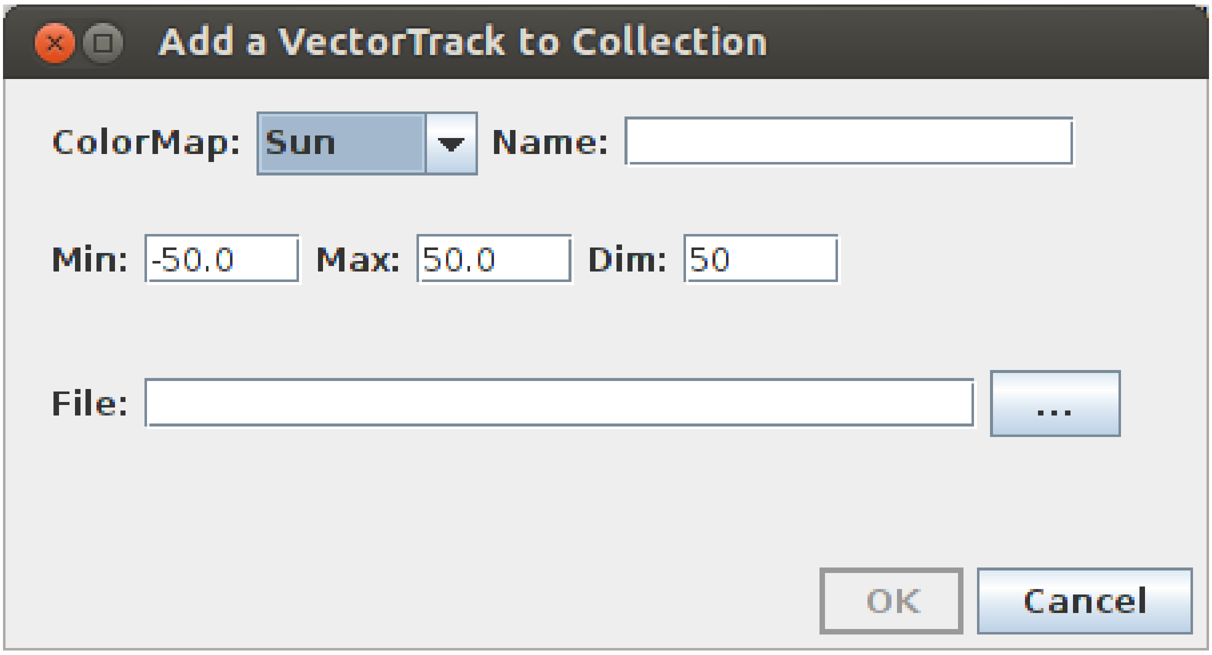
Import Menu

3.4.1. LabelTrack

Import a label-track from a different projects. The imported track will added to the projects folder. They can be found in the projects' 'labeltracks' folders.

3.4.2. VectorTrack

Import a vector-track from raw or external formats. The imported track will be added the project folder. They can be found in the projects' 'datatracks' folders.



ColorMap: Sun
Name:
Min: -50.0
Max: 50.0
Dim: 50
File:

Import Vector-Track

When importing a vector-track several options must be regarded:

Table 3.7. Import Vector-Track options

Option	Use
ColorMap	Atlas provides different color maps to display vector-tracks. Color maps can also be changed later (see LabelArea → Track Display Options).

Option	Use
Name	Provide a name to identify the track.
Min	Minimal value of the used color map. While it may be useful, it is not necessary to choose the minimum value of the track.
Max	Maximal value of the used color map. While it may be useful, it is not necessary to choose the minimum value of the track.
Dim	Dimension of vector data. The vector information for time is not a dimension.
File	The vector track to import.

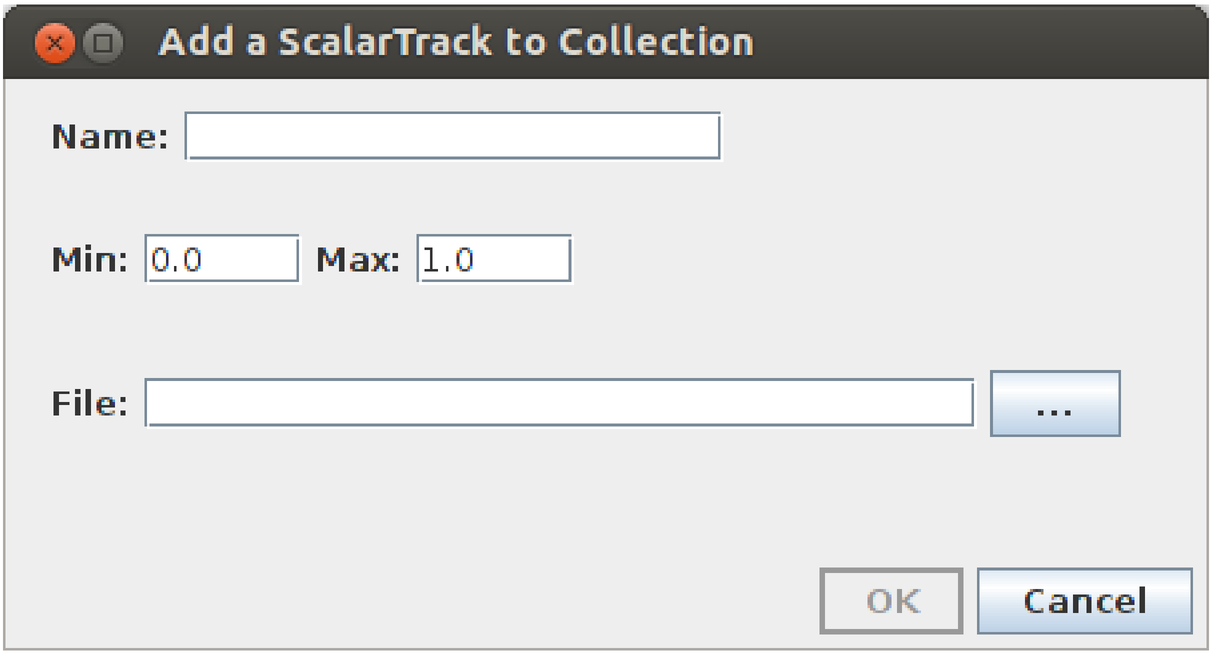
For information about supported files for imports see chapter Atlas Project Files .

3.4.3. Class

Import label-class from other projects. The classes will be added to the project folder. They can be found in the projects' 'classes' folders.

3.4.4. ScalarTrack

Import a scalar-track from raw or external formats. The imported track will be added to the project folder. They can be found in the projects' 'datatracks' folders.



Name:

Min: 0.0 Max: 1.0

File:

Import ScalarTrack

When importing a scalar-track several options must be regarded:

Table 3.8. Import Scalar-Track options

Option	Use
Name	Name used to identify the track.
Min	Minimum value used in the scale to display the track.
Max	Maximum value used in the scale to display the track.
File	Scalar-track to import.

For information about supported files for imports see chapter Atlas Project Files .

3.4.5. Audio

Supported audio formats for import are all audio sources the VLC player can handle. It is highly suggested to only use PCM 16bit WAV format, because Atlas can create a scalar-track from the input file with this format. The user is asked if a scalar-track should be created when importing a WAV file. Using other WAV formats may result in incorrect timelines of the generated scalar-tracks, support for more formats may be added in future releases. Audio conversion to WAV can be accomplished with many audio editing tools depending on operating system and personal preferences.

3.4.6. Video

All video types supported by the VLC media player can be added to a collection using

Import → video.

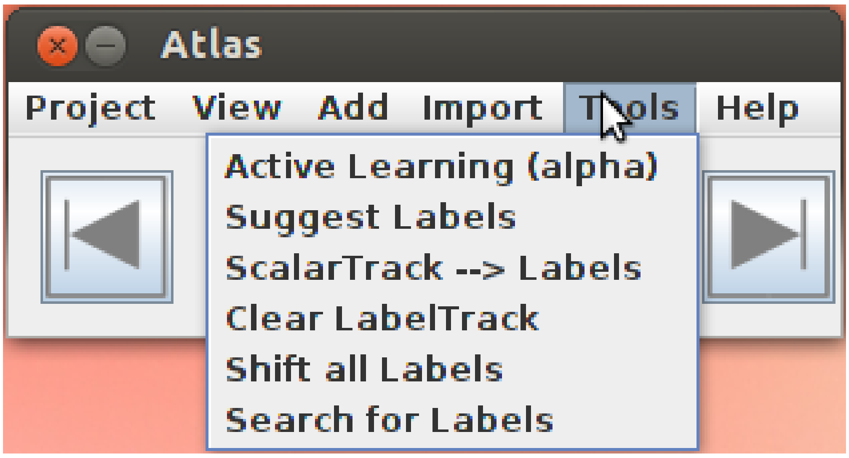
Useful video options are video display size and muted audio (see View → Convenience Options).

Warning

Video imports are not copied to the project folder. If it is preferable to keep all project related input files in the project folder, copy the videos to the subfolder media and import them from there.

3.5. Tools Menu

Atlas offers additional tools for active and semi-supervised learning as well as others to simplify working with Atlas.



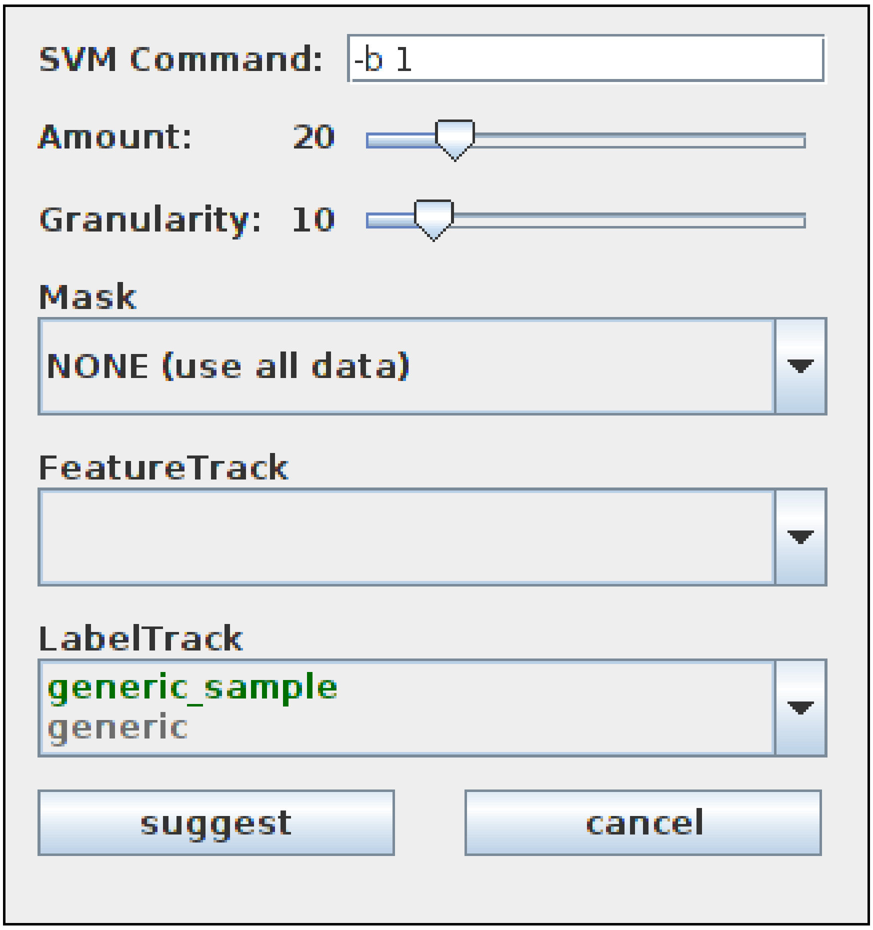
Tools Menu

3.5.1. Active Learning

Currently under development. Do not use. TODO

3.5.2. Suggest Labels

Checks the current project and proposes labels for unlabeled areas of a selected label-track. Atlas uses a data-track as 'feature'-reference for wanted patterns to suggest labels.



SVM Command: -b 1

Amount: 20

Granularity: 10

Mask

NONE (use all data)

FeatureTrack

LabelTrack

generic_sample
generic

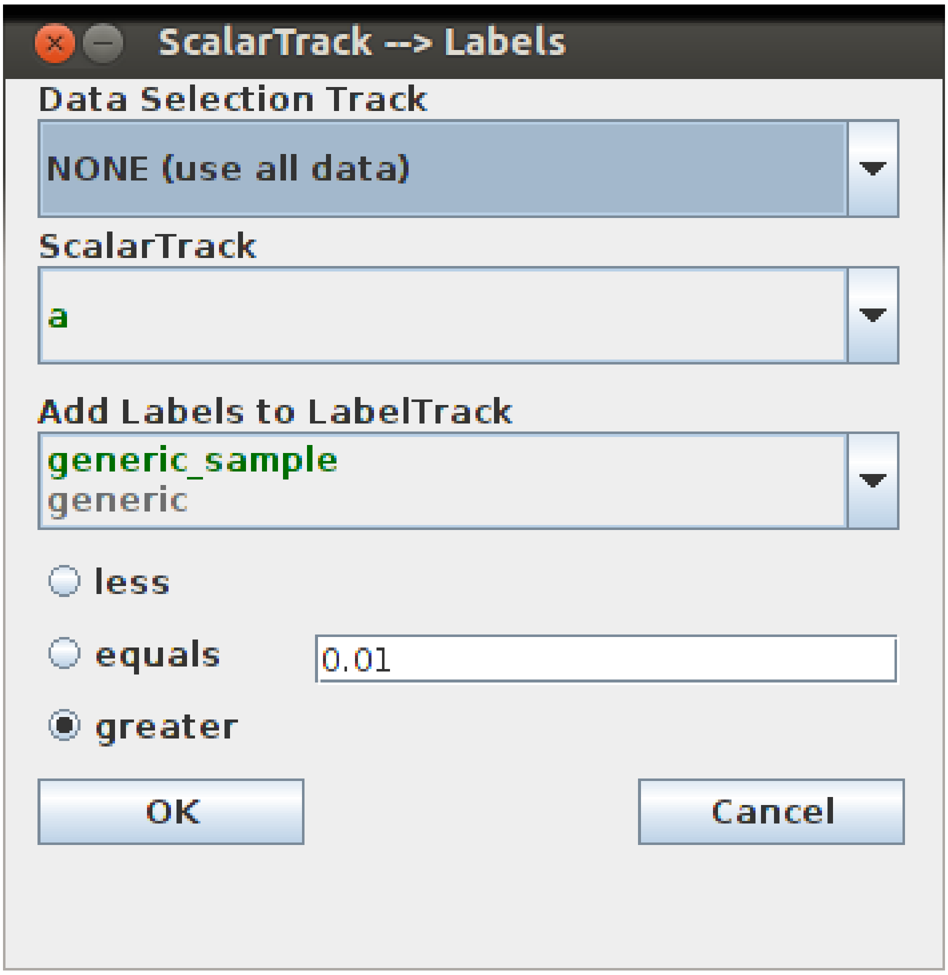
Suggest Labels

Table 3.9. Suggest Labels

Item	Use
SVM Command	The classificator used here is 'libsvm'. Please see *http://www.csie.ntu.edu.tw/~cjlin/libsvm/* for more information.
Amount	Percentage Atlas will try to label of the unlabeled area. The sections Atlas is most confident in will be labeled.
Granularity	Width of analyzed sections in seconds. A label will be proposed for the complete block if applicable.
Mask	A label-track to limit the data of the feature-track for training the classifier that suggests the labels. Create a label-track for masking if areas of the feature-track should not be used for training.
FeatureTrack	The data-track used as feature for auto-labeling.
LabelTrack	The label-track Atlas will try to auto-label.

3.5.3. ScalarTrack --> Labels

This tool allows auto-labeling from the data of a scalar-track. All areas where the data of a scalar-track exceeds a given threshold, a label will be created in the provided label-track.



Data Selection Track: NONE (use all data)

ScalarTrack: a

Add Labels to LabelTrack: generic_sample generic

○ less

○ equals 0.01

● greater

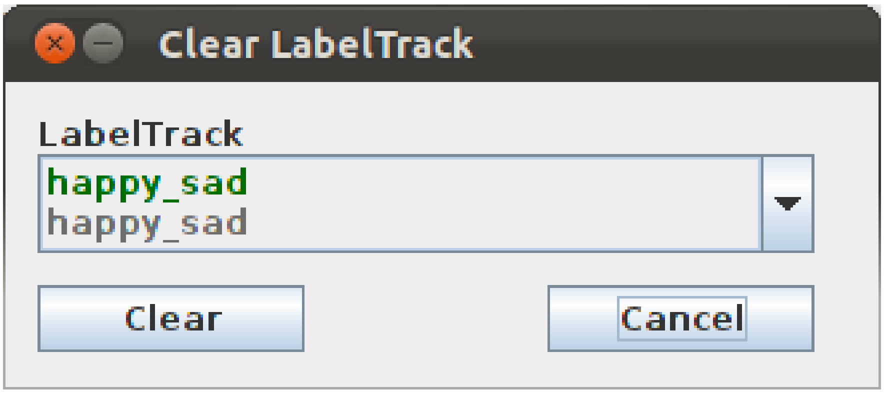
ScalarTrack --> Labels

Table 3.10. ScalarTrack --> Labels

Option	Use
Data Selection Track	A label-track providing a data-mask. Only areas labeled in this track will be examined the auto-labelling. Default is 'NONE (use all data)' which means no restrictions concerning the timeline.
ScalarTrack	The scalar-track used as data input to check against the provided threshold for auto-labeling.
Add Labels to LabelTrack	The label-track the created labels are added to.
less / equals / greater	Label will be created if scalar-track data is less, equal or greater than the provided threshold.

3.5.4. Clear LabelTrack

Removes all labels from a label-track.



Clear LabelTrack

LabelTrack

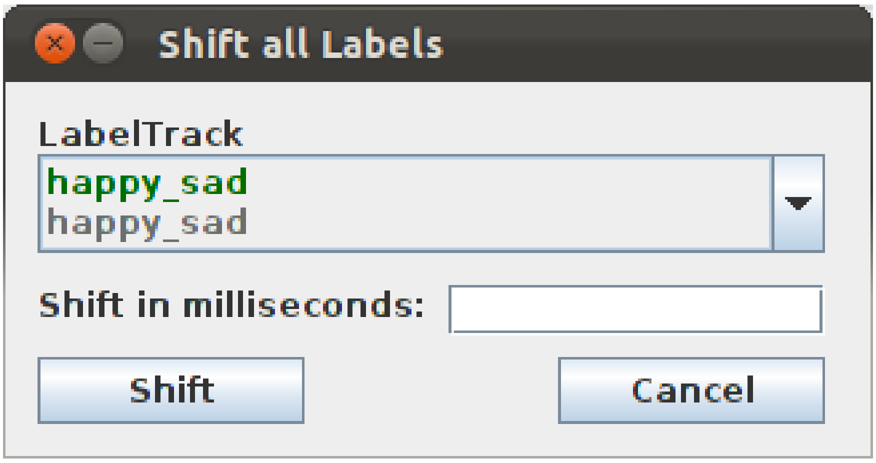
happy_sad

happy_sad

Clear LabelTrack

3.5.5. Shift all Labels

Shifts all labels in a label-track by a provided amount of milliseconds. Negative values shift to the left, postivie values to the right.



Shift all Labels

LabelTrack

happy_sad

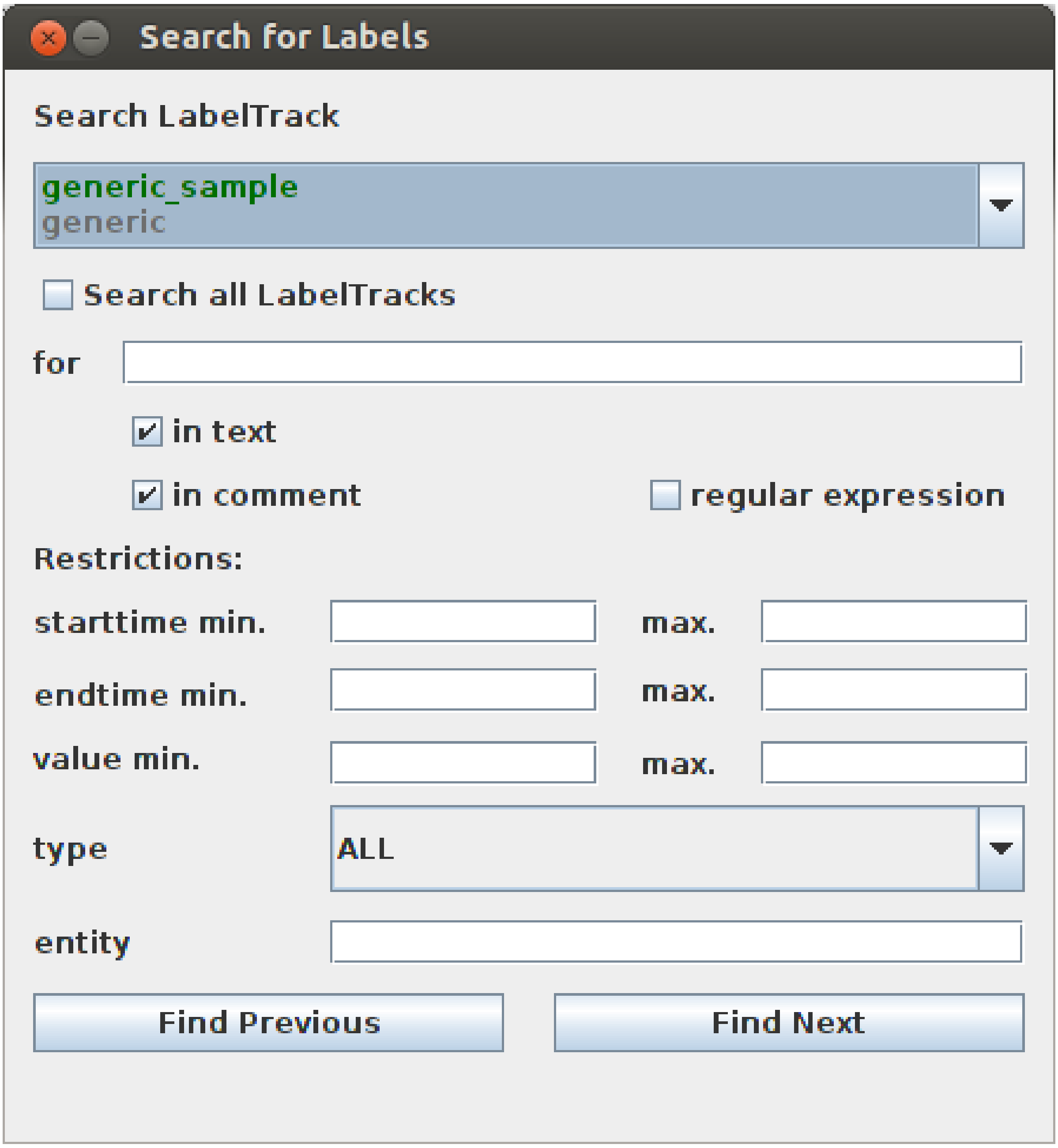
happy_sad

Shift in milliseconds:

Shift all Labels

3.5.6. Search for Labels

This window allows to search labels for certain criteria.



Search LabelTrack: generic_sample generic

☐ Search all LabelTracks

for

☑ in text

☑ in comment

☐ regular expression

type: ALL

Search for Labels The following table lists available options:

Table 3.11. Search for Labels

Option	Use
Search LabelTrack	Choose the label-track that should be searched or select 'Search all LabelTracks' for a global search.
for	Search term to use.
in text / comment	Search the text and / or comment content of labels for matches.
regular expression	Search term is a regular expression.
starttime / endtime	Only lablels with start- and endtime between provided values (in milliseconds) are considered.
value min/max	Only lablels with values between min and max are considered.
type	Select a label type as a search restriction.

Option	Use
entity	The labels entity must match for it to be considered in the search.

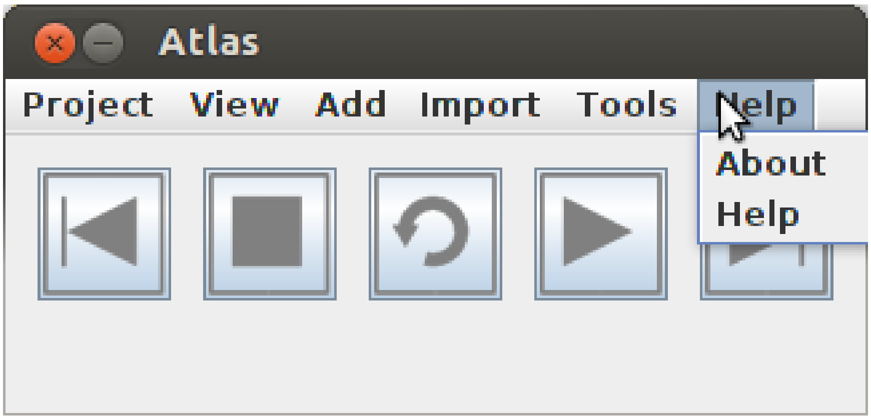
For a label to be considered a match all provided options must be met.

Note

The search always starts at the currently selected label. If no label is selected or the selected label is not in a search label-track, the search will start at the first searchable label ('Find Next') or the last ('Find Previous').

3.6. Help Menu

Atlas also provides help for users:



Help Menu

3.6.1. About

Shows a small dialog with basic information of this software.

3.6.2. Help

Opens a JavaHelp System with the information you are currently viewing.

3.7. Atlas and Files

It is important to note on thing at this point: Atlas never deletes files from the file system. When for example a label-track is created, the label track will be represented as a file in the the project folder. Removing the label-track will NOT delete that file. In order to create a new label-track with the same name, the file needs to be deleted by hand. This applies to all files Atlas creates. A reason for this is the fact that there is no 'undo' function in Atlas. So...

Note

Atlas NEVER deletes a file. When a track is removed and a new of the same type and name, a warning will be shown that the track already exists. Choose a different name, delete the original track from the file system or rename the old track.

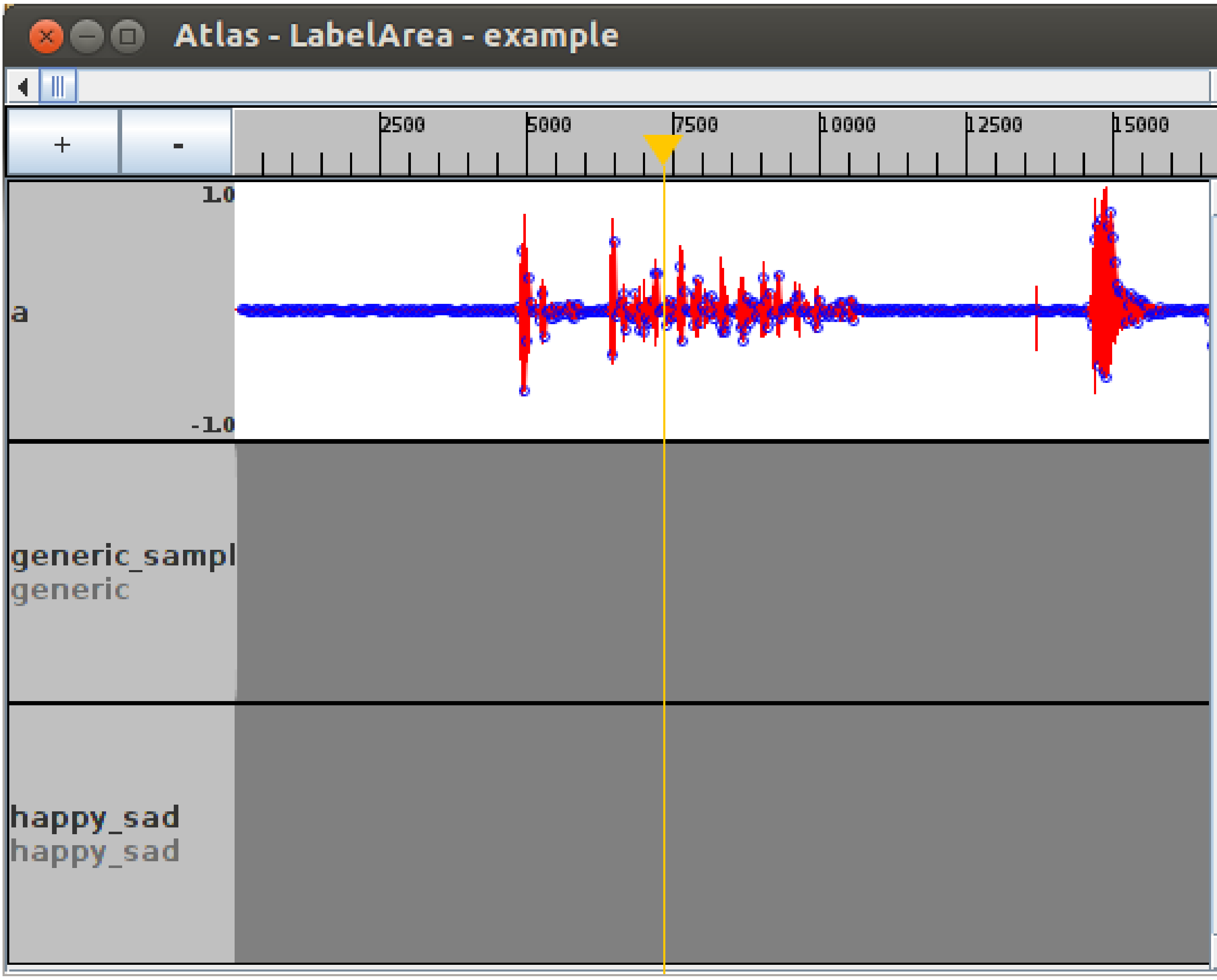
4

The Atlas LabelArea

The LabelArea shows all graphically representable tracks (data-tracks). These include scalar-tracks, vector-tracks and label-tracks with their time-lines. The management of these tracks was shown in chapter Atlas Main Window . This chapter therefore focuses on the graphical interaction options in the LabelArea. General knowledge of tracks, label-classes and entities as described in the previous chapters is assumed. The image below shows the LabelArea of a sample project with a video-track (not displayed in the LabelArea) and an imported audio track named a (wav format). The top track shown in the LabelArea is the scalar-track generated by Atlas from a. The following two tracks are label-tracks. The first one uses the 'generic' label-class with the default entity 'labeled'. The second uses a custom label-class 'happy_sad' with the entities 'happy', 'none' and 'sad' to offer a simple mood label-track.

Note

A label can be added with the mouse by right-clicking at the desired section and selecting 'add'.



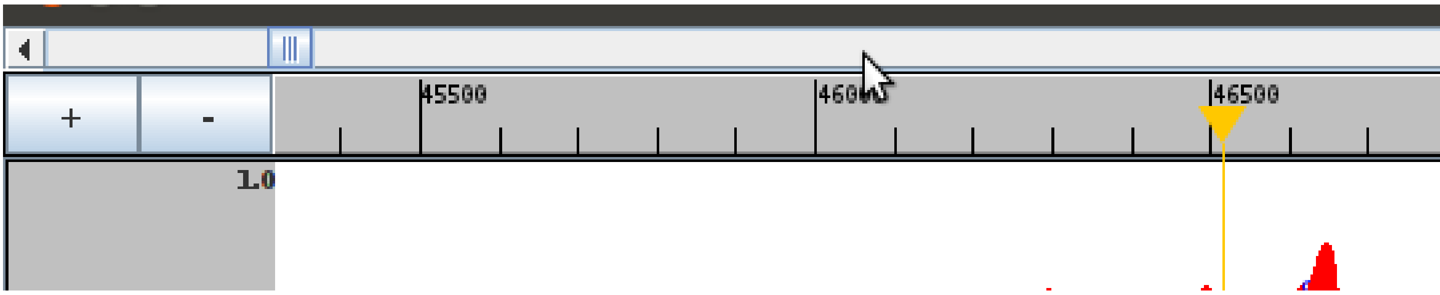
LabelArea

4.1. Time Index, Zooming and Loop Areas

The LabelArea allow selecting points in time and zooming in and out to offer a visual representation of tracks in a desired size. It is also possible to define a 'Loop Area' that can be used to repeatedly play a part of the project.

4.1.1. Time Index and Loop Area

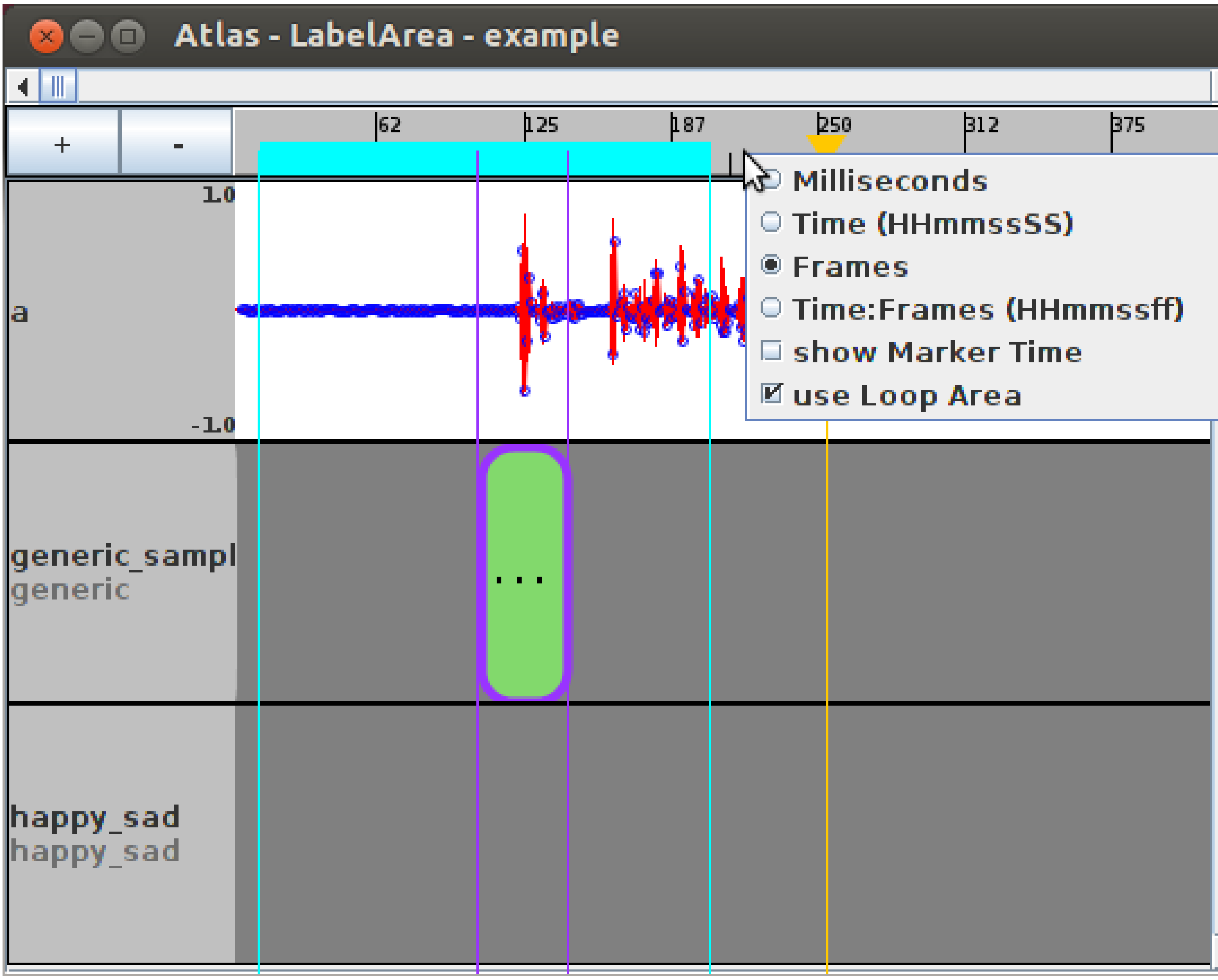
The time index can be changed using the scrollbar at the very top of the LabelArea. The maximum time index is the length of the longest track in the project. Please note that not all tracks need to have the same length. All tracks automatically start at time index 0, delaying a track to start at a different index is not supported.



Mouse over Time Index Scrollbar

The default type of the time index is in milliseconds. It is possible to change this by right-clicking the time index and choosing an alternative format. 'show Marker Time' adds time information to the yellow marker for the current time index while 'use Loop Area' allows for repeated playback of a selected section.

The Loop Area can be adjusted by using Drag & Drop in the time index area. It overrides the loop-playback of the currently selected label.

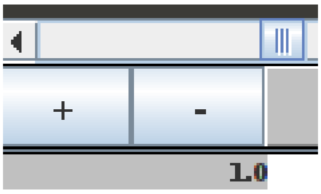


○ Milliseconds
○ Time (HHmmssSS)
● Frames
○ Time:Frames (HHmmssff)
□ show Marker Time
☑ use Loop Area

Change display type of time index, a Loop Area is displayed on the left side.

4.1.2. Zoom Level

Zooming in and out to display tracks in different sizes can be done by using the '+' and '-' buttons at the top left corner of the LabelArea. It is also possible to move the mouse over the tracks listed in the LabelArea and use the mouse wheel to zoom in and out. If a label is selected in the LabelArea, the zooming will home in on the selected label, otherwise Atlas will use the selected time index as reference.



Zoom Buttons

4.2. Managing Labels

Labels can be

- manually added,
- suggested by Atlas (see Tools Menu) or
- imported (see Import Menu).

Labels (including imported labels) are either manually labeled or auto-labeled. Working with manually labeled and auto-labeled labels is slightly different.

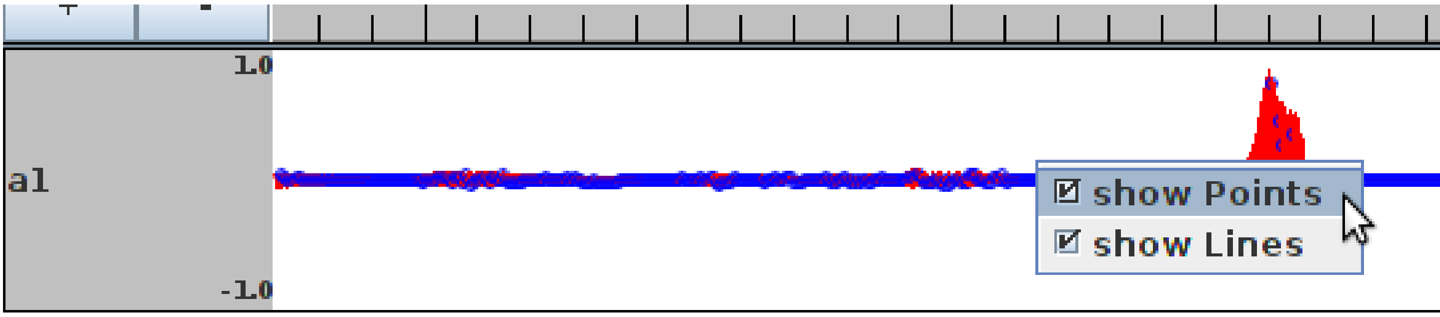
Labels are manually added by right-clicking the respective track in the LabelArea and choosing add. The labels properties can be adjusted using the LabelProperties window.

Using Drag & Drop functionality it is possible to change the start and end point of a label (use left and right border of label for Drag & Drop) or move the complete label as such (use center of label for Drag & Drop). In order to change auto-labeled labels this way, 'CTRL' must be pressed on the keyboard.

Note
Auto-labeled labels can be accepted or rejected in LabelProperties . The LabelProperties window can also be activated by double clicking on a label.

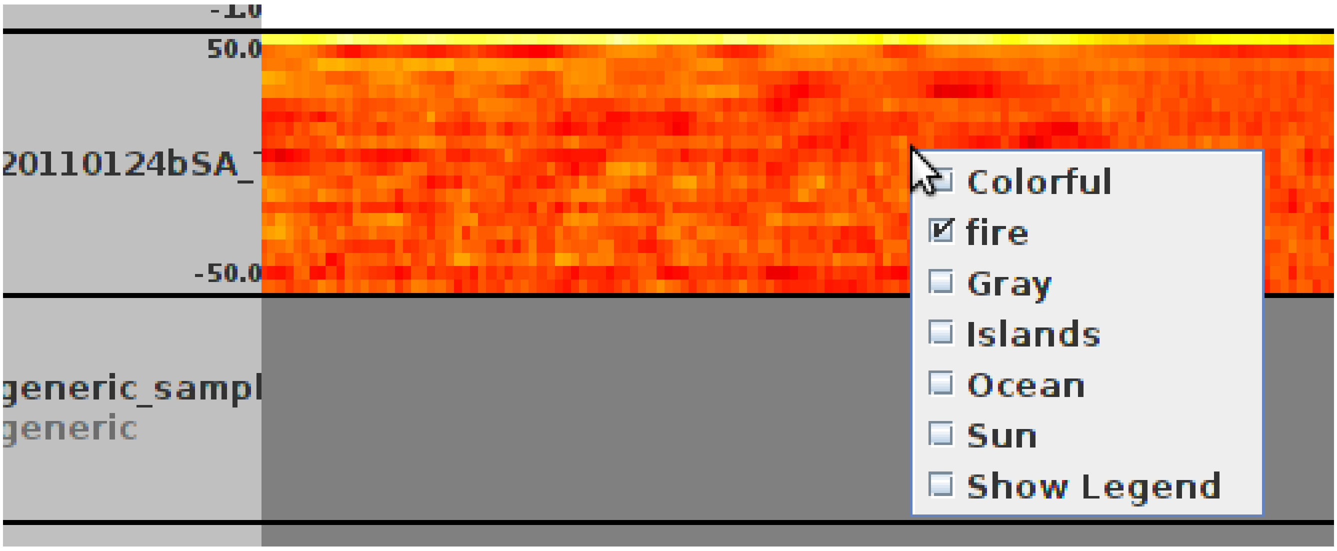
4.3. Track Display Options

Remember that tracks can be managed in the TrackEditor .
Right-clicking on a scalar-track allows to change the display options 'show points' and 'show lines'.



Scalar-Track Display Options

A right-click on a vector-track allows to change the colormap of the vector-track.



Vector-Track Display Colormap change

4.4. Import by Drag & Drop

It is possible to use Drag & Drop to import files directly from the file manager. When needed the respective import dialog will be shown where needed information for the import must be provided.

Possible imports by draging a file from the file manager onto the LabelArea contain:

- label-classes
- label-tracks
- scalar-tracks
- vector-tracks
- audio files
- video files

Not all possible formats of the above inputs may be supported by Drag & Drop, if in doubt simply try to do it.

More information on file formats and folder structure used in Atlas can be found in chapter Atlas Project Files .

5 Atlas Project Files

This chapter introduces structures and files used to organize projects in Atlas. At first the general structure of project folders will be covered followed by specific files themselves. At some point it may be useful to edit some project files by hand. This chapter provides basic information on the files and structures used by Atlas in project management to allow manual editing when required.

Note

Basic understanding of XML files and XML structure is assumed at several sections.

The following sections will describe the project folder's structure as well as the purpose of the subdirectories and the project file <projectname>.xml. Information on data formats supported as imports in Atlas will also be provided in the respective sections.

5.1. Project Folder

When creating a new project a subfolder with the project's name is created in the provided directory. This project folder has the following structure:

Table 5.1. Project Folder

File / Folder	Use / Contains
classes	label-classes
datatracks	scalar- and vector-tracks
labeltracks	label-tracks
media	folder provided as a central location for audio and video files
<projectname>.xml	A file governing the projects state including what tracks are used and how.

5.2. LabelClass Files

LabelClass files are located in the project directory's 'classes' folder. Label-classes have a name and entities. The entities themselves do have name, color and (optionally) id values. The example with the happy_sad class is structured as shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<LabelClass classname="happy_sad">
 <entity>
  <name>happy</name>
  <color>-13312</color>
  <id>1</id>
 </entity>
```

```
  <entity>
   <name>none</name>
   <color>-3355444</color>
   <id>2</id>
  </entity>
  <entity>
   <name>sad</name>
   <color>-52480</color>
   <id>3</id>
  </entity>
 </LabelClass>
```

The LabelClass element representing the LabelClass has the class name and all entity elements. The entity elements consist of name, color and id elements that have the respective value. As mentioned earlier, Atlas does not delete files, therefore all label-classes once used in a project will continue to exist, until manually removed from the file system.

The use of existing label-classes in a project is managed by the project's <projectname>.xml file.

Label-classes can be imported from existing projects by using

Import → Class

and selecting the existing label-class XML file.

5.3. Data-Tracks

The subfolder 'datatracks' contains both scalar- and vector-tracks. First the more general Vector-Tracks are described followed by a section on Scalar-Tracks .

5.3.1. Vector-Tracks

The vector-tracks' 'raw' format is not human readable. Vector-tracks have multiple time-entries. These entries again have data values representing the dimensions of the vector. For a 20 dimensional vector a whole data entry therefore exists of a time entry with 20 data entries.

Import of vector-tracks is possible through

Import → VectorTrack.

or with some cases through Drag & Drop onto the LabelArea.

Possible import formats for vector-tracks are:

Table 5.2. Vector-Track Import Formats

Format	Description
raw	Already existing vector-tracks from other Atlas projects.
XML	The composition of a XML document used for vector-track import is described in XML Import of Vector-Tracks .
CSV	Comma Separated Value import of vector is described in CSV Import of Vector-Tracks .
mat	Matlab provides another way to import vector-tracks, the mat-file is described in Matlab Import of Vector-Tracks .

5.3.1.1. XML Import of Vector-Tracks

The XML file to import vector-tracks needs a root element. This root element then contains 'vector' elements of the form

```
<vector time="100">
 <value>0.4</value>
 <value>0.5</value>
 <value>0.6</value>
</vector>
```

where time is in milliseconds.

An example for an XML file used for vector-track import is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
 <vector time="100">
  <value>0.4</value>
  <value>0.5</value>
  <value>0.6</value>
 </vector>
 <vector time="200">
  <value>0.6</value>
  <value>0.3</value>
  <value>0.6</value>
 </vector>
 <vector time="300">
  <value>0.7</value>
  <value>0.2</value>
  <value>0.5</value>
 </vector>
</root>
```

This file will create a three dimensional vector-track with three vector entries (at 100, 200 and 300 milliseconds).

Warning

The data MUST be ordered in respect to their time value or there will be problems when Atlas uses the vector-track.

5.3.1.2. CSV Import of Vector-Tracks

CSV files can either represent one vector-track or several scalar-tracks. An example is shown below:

```
100,2,0.1
200,3,0,4
300,-0.5,0.3
400,1,-0.2
```

The first column represents the time value. The following columns represent the vector. The example above therefore represents four vectors with dimension two. The following data is imported:

- At 100 milliseconds vector (2,0.1),
- at 200 milliseconds vector (3,0.4),
- at 300 milliseconds vector (-0.5,0.3),

- at 400 milliseconds vector (1,-0.2).

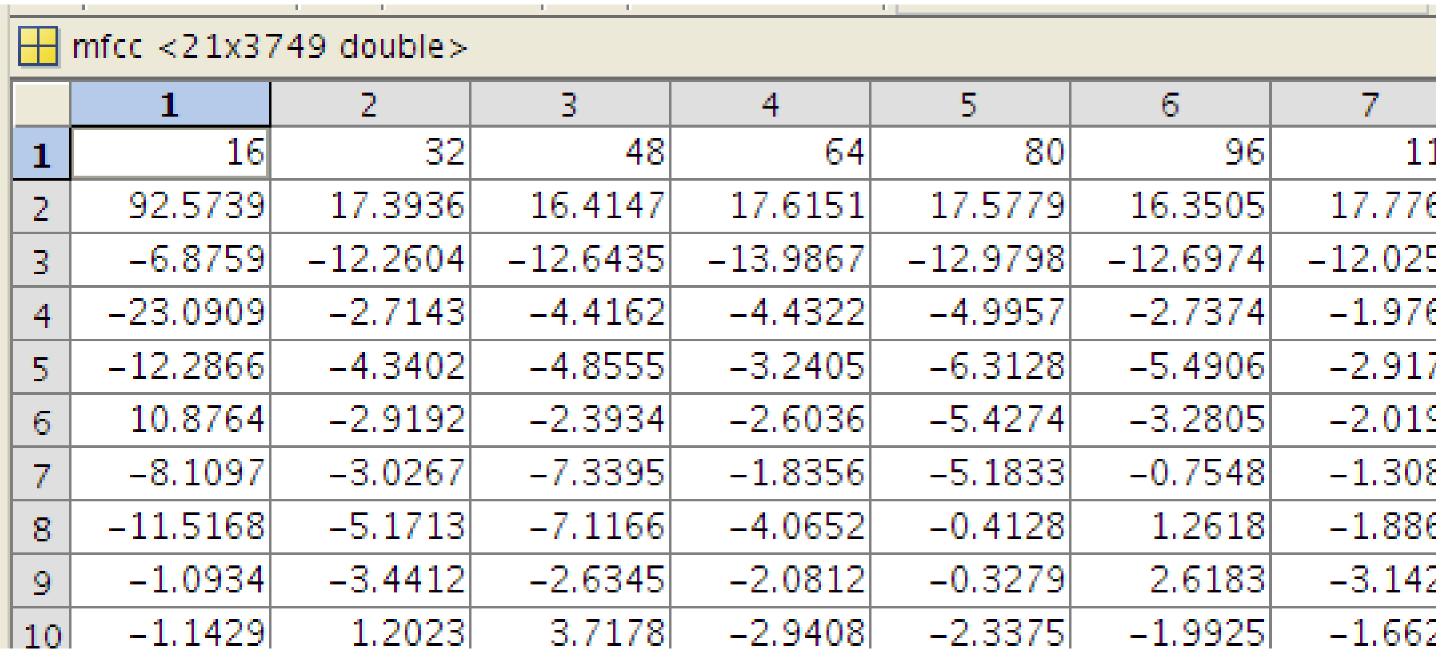
Additional dimensions are represented with additional columns. Columns delimiters are ',', ';' and tabulator. They may be mixed inside a row.

Warning

The data MUST be ordered in respect to their time value or there will be problems when Atlas uses the vector-track.

5.3.1.3. Matlab Import of Vector-Tracks

Atlas allows direct data import from matlab files. The import structure for vector imports is as follows:



mfcc <21x3749 double>

	1	2	3	4	5	6	7
1	16	32	48	64	80	96	11
2	92.5739	17.3936	16.4147	17.6151	17.5779	16.3505	17.776
3	-6.8759	-12.2604	-12.6435	-13.9867	-12.9798	-12.6974	-12.025
4	-23.0909	-2.7143	-4.4162	-4.4322	-4.9957	-2.7374	-1.976
5	-12.2866	-4.3402	-4.8555	-3.2405	-6.3128	-5.4906	-2.917
6	10.8764	-2.9192	-2.3934	-2.6036	-5.4274	-3.2805	-2.019
7	-8.1097	-3.0267	-7.3395	-1.8356	-5.1833	-0.7548	-1.308
8	-11.5168	-5.1713	-7.1166	-4.0652	-0.4128	1.2618	-1.886
9	-1.0934	-3.4412	-2.6345	-2.0812	-0.3279	2.6183	-3.142
10	-1.1429	1.2023	3.7178	-2.9408	-2.3375	-1.9925	-1.662

Matlab Vector-Track Import

The first row represents the time indexes. The vector is in the following rows of the column.

Warning

The data MUST be ordered in respect to their time value or there will be problems when Atlas uses the vector-track.

5.3.2. Scalar-Tracks

Vector-tracks are basically a special form of vector-tracks (vector-tracks with dimension 1). Like vector-tracks scalar-tracks are in the non human-readable format 'raw'. Their data entries consist of time and data values.

Import of scalar-tracks is possible through

Import → ScalarTrack.

or with some cases through Drag & Drop onto the LabelArea.

Possible import formats for scalar-tracks are:

Table 5.3. Scalar-Track Import Formats

Format	HowTo
raw	existing scalar-tracks from other Atlas projects

Format	HowTo
XML	The composition of a XML document used for scalar-track import is described in XML Import of Scalar-Tracks .
CSV	Comma Separated Value import of vector is described in CSV Import of Scalar-Tracks .
WAV	When importing a 16bit PCM mono WAV file, Atlas can automatically create a scalar-track from the input file.
mat	The import of scalar-tracks is described in Matlab Import of Scalar-Tracks .

5.3.2.1. XML Import of Scalar-Tracks

The XML file to import scalar-tracks needs a root element. This root element then contains 'sample' elements of the form

```
<sample time="100">0.5</sample>
```

where time is in milliseconds.

An example for an XML file used for scalar-track import is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
 <sample time="100">0.4</sample>
 <sample time="200">0.6</sample>
 <sample time="300">0.2</sample>
</root>
```

This file will create a scalar-track with three sample entries (at 100, 200 and 300 milliseconds).

Warning

The samples MUST be ordered in respect to their time value or there will be problems when Atlas uses the scalar-track.

5.3.2.2. CSV Import of Scalar-Tracks

CSV files can either represent one vector-track or several scalar-tracks. A scalar-track import is structured as follows:

```
100,2
200,3
300,-0.5
400,1
```

Allowed delimiters are ',', ';' and tabulator. The first columns represents the time indexes the value behind the time indexes the respective data value. It is possible to import more than one scalar-track at once, if the time indexes match:

```
100,2,0.3
200,3,0.4
300,-0.5,0.5
400,1,0.1
```

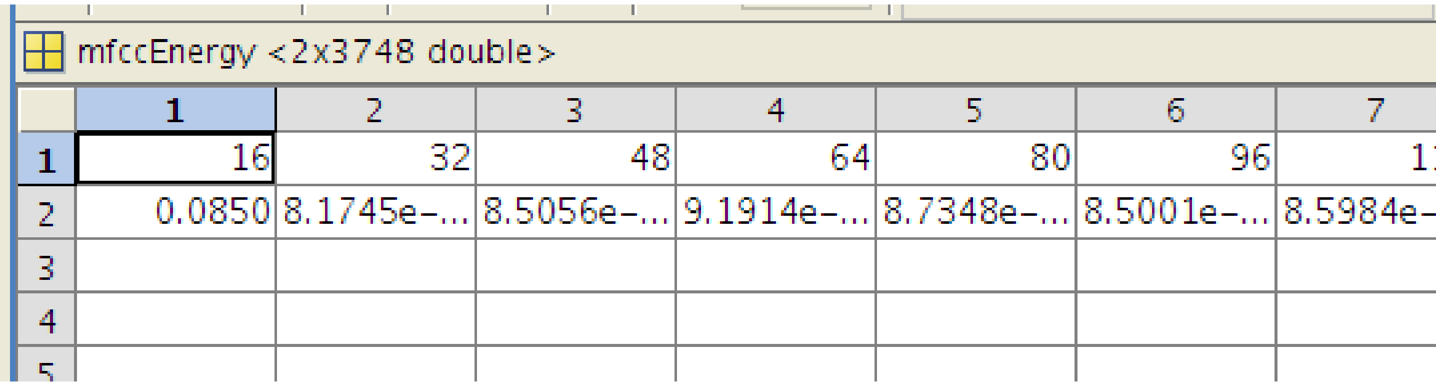
In the example above two scalar-tracks will be imported with the same time indexes. The first one's data is in the second column, the second one's in the third. Atlas will append numbers to the provided name of the scalar track to allow this multi-import.

Warning

The data MUST be ordered in respect to their time value or there will be problems when Atlas uses the scalar-track.

5.3.2.3. Matlab Import of Scalar-Tracks

Atlas allows direct import from Matlab files, an example:



mfccEnergy <2x3748 double>

	1	2	3	4	5	6	7
1	16	32	48	64	80	96	1
2	0.0850	8.1745e-...	8.5056e-...	9.1914e-...	8.7348e-...	8.5001e-...	8.5984e-
3							
4							
5							

Matlab Scalar-Track Import

The first row represents the time indexes of the scalar-track, the values under the time indexes are the respective data.

Warning

The data MUST be ordered in respect to their time value or there will be problems when Atlas uses the scalar-track.

5.4. Label-Tracks

Label-tracks are saved in XML files as shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<LabelTrack name="happy_sad" externalchange="0" classname="happy_sad">
  <label>
    <starttime>247</starttime>
    <endtime>534</endtime>
    <timestamp>1328192148168</timestamp>
    <value>1.0</value>
    <comment>test-comment</comment>
    <type>MANUAL</type>
    <text>label1</text>
    <classentity>happy</classentity>
  </label>
</LabelTrack>
```

This shows the example happy_sad track with a single label. The label-tracks attributes 'name', and 'classname' are obvious, 'externalchange' should be set with the current time in milliseconds.

The attributes are named exactly as described in the LabelProperties window.

It is also possible to import Folker transkrition files. See

http://agd.ids-mannheim.de/html/folker.shtml
for further information.

5.5. Media Folder

The media folder can be used as a central location for the video and audio tracks of a project. When moving a project to a different computer, the paths of video and audio files in the project file <projectname>.xml do not match any more. The same applies to moving or deleting video and audio files. When such a media file can not be found at its specified location, Atlas attempts to load the file from the media directory. While not required, it may be advisable to copy all media files to the the media directory so Atlas will always find the files and projects can be transferred without any adjustments by simply copying from one computer to another.

5.6. Project File - <projectname>.xml

The project file is used to save the state of a project. This section is mainly for enthusiasts. There should be no need to manipulate project files at all. The explanations will therefore be very brief, an simple project's example is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<AnnotationProject name="ttttt" length="1270624.0" zoom="0.6666666666666666">
  <labelclasses>
    <lclass name="generic">generic.xml</lclass>
    <lclass name="happy_sad">happy_sad.xml</lclass>
  </labelclasses>
  <labeltracks>
    <ltrack name="happy_sad" class="happy_sad" active="true" order="2" learnable="false">h
  </labeltracks>
  <datatracks>
    <scalartrack>
      <scatrack name="a1" active="true" type="ScalarTrack" showpoints="true" showline="tru
    </scalartrack>
    <vectortrack>
      <vectrack name="20110124bSA_TBN_E500__ETRG_mfcc" active="true" type="VectorTrack" co
    </vectortrack>
  </datatracks>
  <mediatracks>
    <audio>
      <atrack name="a1" active="true">/home/bautschi/SCHAFFEN/SCHAFFEN/AVEC_003_devel/medi
    </audio>
    <video>
      <vtrack name="v1" active="true">/home/bautschi/SCHAFFEN/SCHAFFEN/Saschatestaufnahme/
    </video>
  </mediatracks>
</AnnotationProject>
```

The root is AnnotationProject which has a name, length (of the longest track in the project) and a zoom factor (video zoom).

The first element is 'labelclasses' and contains all label-classes.

The next element groups the label-tracks. Every single label-track (ltrack element) has a name and label-class (class). The attributes active and order show if the track is visible and its order in the LabelArea. 'learanable' means the 'is feature' attribute.

Next in line are the data-tracks which are divided in scalar- and vector-tracks. Their attributes are named and self explaining, 'order', 'active' and learnable are described above.

The media-tracks divided in audio (with separate atracks) and video (vtracks) do contain information on their file's location in the file system. If a file can not be found there, Atlas will search the projects media folder.

6 Know Issues

This section lists known issues of Atlas that are not (or are not considered as) bugs. Real bugs are not part or this document and a bug management system does not exist at this point.

6.1. libjawt.so not found

This problem was encountered in Ubuntu 12.04 in combination with Java Openjdk 7. The LD_LIBRARY_PATH must contain the path to the libjawt.so file. To resolve this the file must be located on the file system and its parent folder be added to LD_LIBRARY_PATH. This is best done in a startup script without trying to modify the global system environment. The script could look something like:

```
#!/bin/sh

# add path of libjawt.so to LD_LIBRARY_PATH
LD_LIBRARY_PATH=<PATH TO PARENT FOLDER OF libjawt.so>:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH

# start Atlas
java -jar Atlas<version>.jar
```

Using the path of libjawt.so as found on the Ubuntu 12.04 test system the startup script would be:

```
#!/bin/sh

# add path of libjawt.so to LD_LIBRARY_PATH
LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk-amd64/jre/lib/amd64:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH

# start Atlas
java -jar Atlas<version>.jar
```

6.2. Video Imports General Guidelines

Atlas uses the Video Lan Client (VLC) for video and audio playback, therefore Atlas must use the native VLC libraries. Incompatibilities may occur between Atlas and the VLC libraries. Another problem are not matching JNA files. This section provides a starting point to approach such problems and lists known combinations of operating systems and VLC versions.

Note

If at all possible try to use at least VLC version 2.0 or greater and the matching Atlas version. The Atlas version for VLC 1.x is considered legacy software, will be dropped and has more problems with operating system and Java compatibility.

6.2.1. VLC is too old (only concerns VLC 1.x legacy Atlas)

The implementation of VLC changed at some point that makes Atlas unable to use VLC libraries older than a certain version. A problem is that the exact version number that starts to work is unknown. Versions newer and including 1.1.11 are recent enough.

Note

If at all possible try to use at least VLC version 2.0 or greater and the matching Atlas version. The Atlas version for VLC 1.x is considered legacy software, will be dropped and has more problems with operating system and Java compatibility.

6.2.2. 32bit and 64bit Systems

There may be problems when mixing 64bit operating systems, 64bit java virtual machines and 32 bit VLC. At this point there are systems in both Ubuntu and Windows 7 that had no problem with that combination, but most did not work. The reason for this inconsistancy is unclear.

Use the appropriate VLC and Java versions matching your operating system architecture as mixing 32 and 64 bit will most likely result in incompatibilities. For Windows 64 bit builds of VLC are available at the official VLC website.

Note

If at all possible try to use at least VLC version 2.0 or greater and the matching Atlas version. The Atlas version for VLC 1.x is considered legacy software, will be dropped and has more problems with operating system and Java compatibility.

6.3. VLC stops if its close button is clicked

This issue will not be fixed.

Note

When VLC is terminated by clicking the close button on its window, the track itself is still part of the project. Do not try to close VLC windows. Use the TrackEditor to set videos invisible or to remove them from a project respectively.

Clicking the close button will have no impact on the project file. Therefore a project can simply be saved and reloaded after accidentally clicking a VLC window's close button.

6.4. (Un)Successfully tested Operating System, Java and VLC Versions

Table 6.1. Import VectorTrack options

Success	OS	Java	VLC	Atlas version
maybe *	Windows 7 64bit	1.6 64bit	1.1.11	legacy
yes	Windows 7 64bit	1.6 or 1.7 64bit	2.x	current
maybe *	Ubuntu 11.04 64bit	openjdk-6	1.1.11	legacy
yes	Ubuntu 11.10 32bit	openjdk-6	1.1.11	legacy
yes	Ubuntu 12.04 64bit	openjdk-6 or openjdk-7	2.0.1	current
no	Ubuntu 11.10 64bit	openjdk-6	1.1.11	legacy
no	Mac OS	unknown	1.x	legacy
yes	Mac OS	unknown	2.x	current

* - tested with mixed results, unclear why

If you use Atlas with other systems please notify us of your findings through sascha.meudt@uni-ulm.de.