



## What is Videograph?



### Overview

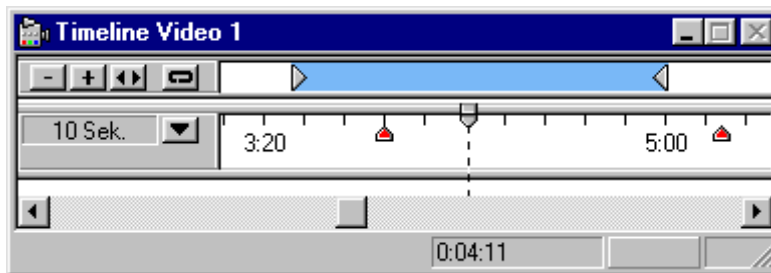
Videograph® is a Multimedia Player with which digitalized videos, e.g. recordings of classroom instruction, can be played and at the same time be evaluated ("videographed"). The program enables the construction of observation categories and rating scales which the viewer can use as a "measuring instrument" to analyze the contents of the video. The coding can take place synchronously while the film is running. It can be segmented in time intervals ("time sampling") or refer to what happens ("event sampling"). The data created are graphically presented on the screen and can be transferred for statistical calculations or a graphic presentation to an external file, whereby the user can choose between the file format of the SPSS statistic program or a tab-delimited text format to import in a word processing program or in spreadsheet programs like Excel. Transcriptions of the spoken contents can also be made. These transcripts can also be exported.











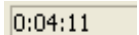


### Features

- All of the Videograph working space elements can be arranged at will. At the end of a session the program memorizes the status of all of the important parameters so when the program is restarted the same working environment is restored.
- All of the functions for steering the video clip are joined in the timeline window and are easy to handle. A time ruler with a movable slider as pointer and a time unit pop-up menu for choosing time increment show the current time position to the exact second. You can navigate exactly, quickly and flexibly. You can also set markers and choose playback intervals.
- Depending on the system capacity, several clips can be played at the same time. It is possible to couple the clips so that one clip is a master clip that provides for exact synchronization of the other clips.
- The program has a comprehensive and user-friendly instrument to code videos. The coding procedure is limited by almost no restrictions and the procedures are flexible so that it is possible, e.g., to quickly redefine, modify or delete coding variables within the program while it is running. Codings can also be revised or renewed at will when necessary. A special strength of the program is the clear and flexible arrangement of the elements as well as the synchronous graphic visualization of the coding data in the form of a time diagram.
- Only a few clicks of the mouse are needed to export data and for the most extent it happens automatically. The little user information the program needs is easily acquired in the dialogue; the rest - including creating data structures - is taken care of by the program.

## ► View of the timeline window



-  Moves the playback area backwards one length
-  Moves the playback area forward one length
-  Locks the two markers of the playback interval. It may be moved, but his duration remains fixed. Serves as protection against unintentional relocations in the course of coding and transcribing, e.g. by shifting the time increment ( important: see under securing playback intervals).
-  Click the loop button to play the clip in a continuous loop. To stop the action, press the spacebar or click inside the video window. Works only when playback interval is activated.
-  Time unit menu: Click triangle to read or set the time base, e.g. the time increment between two tick marks in the time ruler ( e.g. 10 or 20 seconds)
-  Moveable jump marker. Markers provide a way to indicate important points in time. They help you position a clip. To go to a marker you just need to push a key. You can annnotate markers also.
-  Moveable jump marker. Markers provide a way to indicate important points in time. They help you position a clip. To go to a marker you just need to push a key. You can annnotate markers also.
-  The **playback interval** is the area between the two markers. When the area is activated (to be seen from blue color), only the section can be played which is within this interval. A playback interval is activated and deactivated by double clicking a position somewhere between the two markers with the left mouse key .
-  The marker on the left edge sets the beginning of the playback interval. You can position the marker using the mouse
-  The marker on the right edge sets the ending point of the playback interval. You can position the marker using the mouse
-  Shows exact time position (h:mm:ss) of the clip during playing or when the pointer is relocated.

## ▶ Controlling the video clip in the timeline window

### Scrolling the time ruler

Depending on how long the video clip is and the time increment decided upon (see set time unit), the time ruler is usually not able to visualize the entire time span, but only a more or less long section. In order to show other sections of time, use the scrollbar of the timeline window. By pulling the thumb the time ruler can be „pushed through“ from beginning to end (whereby the time position of the video does, of course, not change, only the section of time shown in the ruler).

## ▶ Controlling the video clip in the timeline window

### Setting the time unit

Using time unit menu the **interval** between two tick marks on the time ruler can be pinpointed. The time span visible in the ruler is, thus, variable and can be adapted to current needs. The available raster lengths are 1,2,5,10,20,60 seconds and "all".

Please **pay attention** to the comments about changing the time unit found under [securing playback intervals](#).

## ▶ Controlling the video clip in the timeline window

### Fast forwarding the clip

- Hold the slider with the mouse with the left mouse key pressed down and drag it to the right until the desired position in the time ruler has been reached. The ruler scrolls automatically further for positions outside the visible area of the window once it touches the right hand edge of the window. Then release the left mouse key.  
**or:**  
press the key CURSOR RIGHT repeatedly. Every time the key is pressed moves the slider one second forward.
- When the SHIFT key is pressed when fast forwarding, the steps made by the slider depend on the set time increment, i.e. it always moves forward according to full tick marks. It is, thus, possible to realize fast forwarding or rewinding at various speeds.
- When forwarding, the pictures are shown which means you do not have to feel your way forward. The pre-requisite is, however, that the speed is not higher than the system can handle.
- By the way: Whenever you move the mouse along the time ruler with the left mouse key pressed - even without moving the position pointer - a little information window „travels“ too, indicating the current position to the second of the mouse pointer.

## ▶ Controlling the video clip in the timeline window

### Rewinding the clip

- Hold the slider with the mouse keeping the left mouse key pressed and pull it to the left until the desired position in the time ruler has been reached. For positions outside the visible area of the window the ruler automatically scrolls further as soon as the slider touches the left edge of the window. Then release the left mouse key.  
**or:**  
repeatedly press the CURSOR LEFT key. Every time the key is pressed the slider moves back one second.
- When the SHIFT key is pressed while rewinding, the steps made by the pointer depend on the set time increment, i.e. it always equals the distance between tick marks. In this way it is possible to realize rewinding at various speeds.
- When rewinding, the pictures are shown which means you do not have to feel your way forward. The pre-requisite is, however, that the speed is not higher than the system can handle.
- By the way: Whenever you move the mouse along the time ruler with the left mouse key pressed - even without moving the position pointer - a little information window „travels“ too, indicating the current position to the second of the mouse pointer.

## ▶ Controlling the video clip in the timeline window

### Going to a time position

The position pointer and, thus, the video clip can be relocated direct to any other time position by double clicking the position where the pointer should be relocated with the left mouse key. When necessary you can move the time ruler with the scroll bar of the timeline window until the desired position is visible. It is not necessary to head for a certain position in the clip by spooling.

When the SHIFT key is pressed during the double click, the slide moves to the next tick mark, i.e. the possible time positions depend on the set time unit.

## ▶ Controlling the video clip in the timeline window

### Replay/Pause

In order to start the replay, press the space key. To pause the clip, press the space key again.

Warning: under Windows only the „activated“ window transmits input from keys, i.e. the function only works when the timeline window is activated (a window can be activated at any time when the mouse is clicked on title bar or on any point inside).



## ▶ Controlling the video clip in the timeline window

### Setting jump markers

Jump markers are there to mark the time position of pictures or video scenes. They can be used to select sections of the video so they can be easily found later. They can be set, relocated, deleted, be annotated and reached at any time - you just need to push a key. All of them are stored at the end of the program; thus, permanently available. There can be any desired number of them.

- In order to set a jump marker:
  - Using the mouse point at the desired position in the time ruler. Press the "M" key and then the left mouse key, or:
  - Press the key combination CTRL+"M" and the jump marker appears in the current time position.
- In order to delete a jump marker:  
Point the mouse to the jump marker. Press the DEL key and then the left mouse key.
- In order to relocate a jump marker:  
Hold the jump marker with the mouse keeping the left mouse key pressed and pull.
- In order to annotate a jump marker:  
Click on the marker using the right mouse key. A small annotation window appears with a blinking caret. Here you can enter text (maximum 255 characters). Click anywhere outside the annotation window to close it. To make the annotation visible again, to alter it or to delete it, click on the marker again with the right mouse key. Deleting the marker also means the annotation is deleted.
- In order to replay the clip from a marker position on: Using the PLUS and MINUS keys in the numerical part of the keyboard, the jump markers can be reached one after the other. Press the PLUS key once to reach the next jump marker (in reference to the current time position) or press the MINUS key once to back to the previous one. Press PLUS or MINUS twice to get to the next but one marker, etc. The same effect is reached with the key combinations CTRL + "+" resp. CTRL + "-".
- When the loop mode is activated (see further below), only those markers within the loop interval can be reached.
- To save notes and time positions of the jump markers to a file: Choose the command "Export as text (jump marker)" in the menu „Import/Export" of the timeline window.

## ▶ Selecting areas to be played

### Overview

Sometimes the replaying of a clip is to begin at position A and automatically end when position B is reached. When coding lesson periods in fixed time intervals it is often important that one does not have to worry about counting units of time or about stopping the film at the end of an interval. In Videograph you can enter the **playback interval**, which specifies the portion of the clip, that you want to play, i.e. starting point A and end point B. Under these conditions only sections can be played which are within this interval. Reason: When the position pointer is outside the selected interval before beginning the replay, replay always begins at position A. When replay begins within the interval, it begins at the current position. In both cases replay stops automatically when position B is reached.

The definition of the boundaries (and length) of a playback interval takes place with two **interval markers**. These are completely flexible and can be set or relocated at will. The playback interval can be moved forward and backward, locked or set in loop mode at will using buttons.

## ▶ Selecting areas to be played

### Setting interval markers

The interval markers are located in the bar above the time ruler. There is a marker on the left edge that sets the beginning of the playback interval and one on the right edge that marks the end of the interval.

- 1. In order to **position** the marker on the **left**:  
Hold the marker with the mouse with the left key pressed down and pull it to the right or the left to the desired time position. Use the time ruler as orientation or read the current position of the mouse from the time indicator which is constantly visible while moving the marker. The shortest move of the marker is 1 second unless the SHIFT key is pressed while moving the marker. If it is pressed, the marker can only be moved from one tick mark to the next, i.e. the available placement marks depend on the chosen time increment.
- 2. In order to **position** the marker on the **right**:  
The right marker can be moved like the left one, viz. by holding and pulling with the mouse to the desired position. There is, however, another, directer way. The marker is placed by simply moving the mouse pointer to the place in the marker bar where the desired time position is and the **right** mouse key is clicked. If necessary, first scroll the time ruler to the right area, viz. when the current section of the ruler does not show the placement point.  
The shortest step that can be made with the marker is 1 second, unless the SHIFT key is pressed while moving. If it is pressed, the marker can only be moved from one tick mark to the next, i.e. the available placement marks depend on the chosen time increment.
- 3. When the playback interval is altered, every marker has to be reset should the old and the new interval not overlap. When there is a larger time span between the old and the new interval it can happen (depending on the set time unit) that the time ruler has to be scrolled whereby the markers go out of sight. In this case there is a fast method to transport **both markers** in the visible section of the ruler at the same time: When you double click somewhere on the ruler with the **left mouse key** the left marker automatically moves to the left edge of the ruler and the right one to the right edge from where you can use the methods described in 1 and 2 to move the markers to their final positions.  
When the new playback interval lies completely within the old one, the **case is different**. For reasons explained in "Toggle playback interval on/off" the double click method does not work here. Therefore, the following tip: Retrieve the **right hand marker** with a click of the right mouse key and select a position which leaves room on the right hand edge of the ruler. Then double click a position between the right marker and the edge of the right hand ruler.
- Another method of relocating both markers at the same time by a constant amount is described in the section "shifting playback interval".
- **By the way**: whenever you relocate a marker, a little information window "moves" as well which means you can read off the momentary position exact to the second.

## ▶ Selecting areas to be played

### Toggle playback intervals on and off

After the playback interval has been defined by positioning the two interval markers it has to be activated, i.e. it has to be "switched on" - otherwise it will not work. A playback interval is activated by **double clicking** a position somewhere along the section between the two markers with the **left mouse key**. The playback interval always assumes a **blue color** when it is activated. The blue disappears when it is switched off. To deactivate follow the same procedure as to activate

## ▶ Selecting areas to be played


### Shifting the playback interval

The playback interval can be enlarged, reduced, or moved to another time position by relocating the two interval markers accordingly. If the playback interval is to be moved to consecutive time sections, the use of the buttons "**move playback interval forward**" and "**move playback interval backward**" (see below) is recommended to move the interval automatically forward or backward by its length, i.e. the beginning of the new interval is joined to the end of the old one or the new interval end with the beginning of the old one.

The playback interval can also be relocated to the **beginning of the clip** when the "move playback interval backward" button is clicked while the **SHIFT key is pressed**.

- **By the way:** when you leave the mouse key stand for a moment, a little information window shows the beginning and end time of the interval.

 Move playback interval forward

 Move playback interval backwards

## ▶ Selecting areas to be played

### Play looped

Using the "Loop"-button, the portion within the **activated** playback interval can be repeated endlessly until it is stopped. The break between 2 runs lasts 1 second.

This function is, e.g. useful when a transcription is to be made of the verbal statements made in the clip.

## ▶ Selecting areas to be played

### Securing playback intervals

Using the "lock playback area" button changes in the length of the interval chosen can be blocked. The interval markers cannot be relocated individually. The interval can only be relocated as a whole by using the buttons "move playback area forward" and "move playback areal back". There are two possible reasons for such a measure:

- Accidental manipulation of the interval markers is hindered.
- **The other reason:** the shortest length that can be set for a playback interval is the time between 2 tick marks on the time ruler. Put in other words, the two interval markers can never be closer than the distance between to tick marks because it would cause optical confusion. This means, e.g., that when the time raster is 20 seconds it is not possible to define a playback interval of less than 20 seconds. It follows that a 20 second interval is transformed into a 1 minute interval when the time increment is altered to 1 minute which is usually undesirable. When the "lock playback interval" button is activated you can only switch to a time unit that is greater as the current length of the interval.

## ► Tips and hints on the left marker

### Left marker is hidden from view

An example: the time between the 15th and the 18th minute is shown in the time guideline. You want to set the left marker at 16:30 but the marker is not visible as it is currently set at 10:30. There are two possibilities:

- Scroll the time ruler to where the marker is and proceed as described in "placement not visible".
- The second possibility is usually easier. When you double click the left mouse key somewhere on the ruler, the left marker automatically jumps to the left edge of the ruler (in our example at position 15:00), and the right marker jumps to the right edge of the ruler (position 18:00). Now you just have to take hold of the left marker with the mouse (holding the left mouse key pressed) and pull it to the right place.

In the example this method only functions when the position of the double click is not within the area enclosed by the two markers at the time of the double click (reason: see "toggle playback interval on and off").



## ▶ Tips and hints on left marker

### Goal Position not visible

Should the position you want to pull the marker to currently be outside the visible part of the time ruler, you have three choices

- Scroll the time ruler until the goal position is visible, best using the left edge of the ruler window. When you use the left mouse key to double click anywhere on the marker bar, the left marker automatically jumps to the left edge and right one to the right. Now, if necessary, you can move the left marker to its final position as well as the right one (see above). This method is only suitable when the place where you double click is outside the area enclosed by the two markers (reason: see " ;toggle playback interval on and off").
- Scroll the time ruler in the direction of the goal position, but only as far as the left marker remains visible. Pull the marker to the edge of the guideline and repeat this procedure until the goal position is visible.
- Switch to a suitable time increment.

### Miscellaneous

The left marker can, of course, not be behind the right one. When the beginning of the new playback interval comes after the end of the previous one, the right marker has to be positioned first.

## ► Tips and hints on right marker

### Right marker is hidden from view

As the right marker can be directly relocated anywhere with a click of the right mouse key, this is the ideal solution for cases when the right marker is not visible because it is in a time position which is not shown in the current section of the guideline. (Note: since the left mouse key is needed for other things, the analogous procedure for the left interval marker cannot be used.)


### Goal position not visible

Since the right marker can be relocated anywhere with a click of the right mouse key, this problem is easily solved: Scroll the time ruler until the goal position is visible and click on the marker bar with the right mouse key to place the marker here.

### Miscellaneous

The right marker, of course, cannot be before the left one. When the end of the new playback interval lies before the beginning of the previous one, the left marker has to be placed first.

► **Keyboard shortcuts in the timeline window**

Key(n)	action
SPACEBAR	Play/Pause
CURSOR DOWN	scroll databars upward
CURSOR UP	scroll databars downward
click left mouse key while "M" key is pressed	Set a jump label at the current position of mouse pointer in the time ruler
click left mouse key while <b>DEL</b> key is pressed	Delete jump label at the place where the mouse pointer is
CURSOR RIGHT	wind forward clip by 1 second
CURSOR LEFT	rewind clip by 1 second
CURSOR RIGHT and SHIFT	wind forward clip to next tick mark
CURSOR LEFT and SHIFT	rewind clip to next tick mark
SHIFT- key with right clicking on the head of databars	Fade-in variable label
only if there is an <b>active</b> databar:	
TABULATOR	move up following databar
TABULATOR+SHIFT	move down active data bar
otherwise:	
TABULATOR-key	activates the coding - and transcript window
only if playback interval is <b>activated</b> :	
HOME (POS 1)	Play from start of interval
SHIFT and mouse clicking 	shift playback interval to start position 0:00:00
only if <b>jump markers(s)</b> set:	
NUMPAD*-	move clip to preceding jump label
NUMPAD*+	move clip to proximate jump label

\*NUMPAD = numeric keypad

## ▶ Workspace

### Project file

An important base for working with Videograph is the project file. A project consists of all steps and outcomes belonging to the coding and transcribing of a certain media clip. A project file therefore contains the complete coding data and all of transcription text as well as a lot of project settings informations including the name of media clip. In order to write a project file you have to select the command "**Save As**" in the file menu and to specify a file name. The names of Videograph-files always end with .vdg, e.g. "project 1.vdg". This ending is appended to the choosed name automatically by Videograph. In order to save a already loaded project file under the same name you can take the simple command "**Save**". In order to open an existing project file you have to use the command "**Open**".

Every working session with Videograph **starts** with the choice of a project file, either by reopening any existing or by creating a new project using the command "**New**" of the file menu, in that case you must also specify a new media clip. And every working session **ends** with the securing of data by saving the project file. In order to avoid loss of data, Videograph will remember you before exiting when you have forgotten to save.

## ▶ Working with windows in Videograph

### Video window and timeline window

When starting the program a timeline window is the only work space component which shows no function and only serves as a place saver. The effective elements are created when a video clip is opened or a project file is opened.

#### The video window

The video window is the medium for replaying a video clip. Every video clip has its own video window which is created when the clip is opened. A video window can be infinitely enlarged or minimized. You will find exact instructions about handling video clips [here](#).

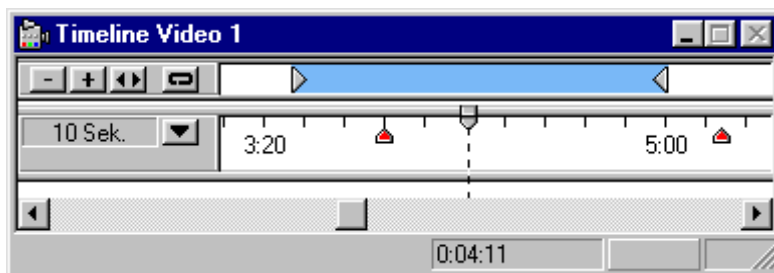
#### The timeline window

All the elements for controlling the video clip are found in the timeline window. When a clip is coded, the data are sent to the timeline window and shown there in graphic form.

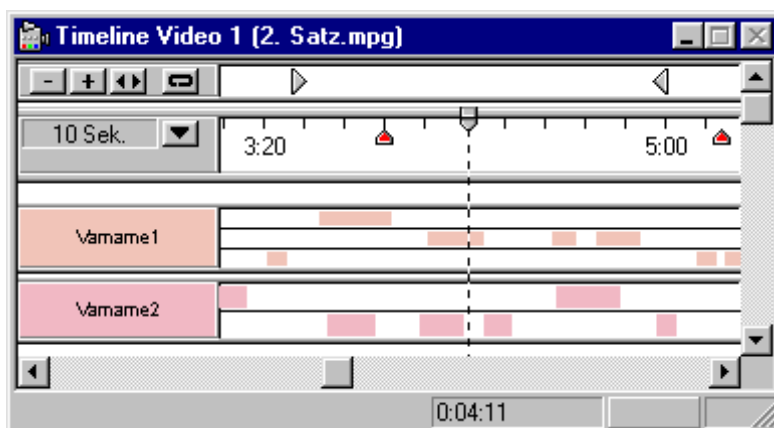
Every video clip has its own timeline window that is created upon opening the clip.

A detailed description of the function and handling of timeline window is to be found [elsewhere](#).

#### ▶ View of a timeline window without data graph



#### ▶ View of a timeline window with data graph



## ▶ Working with windows in Videograph

### Coding and transcription window

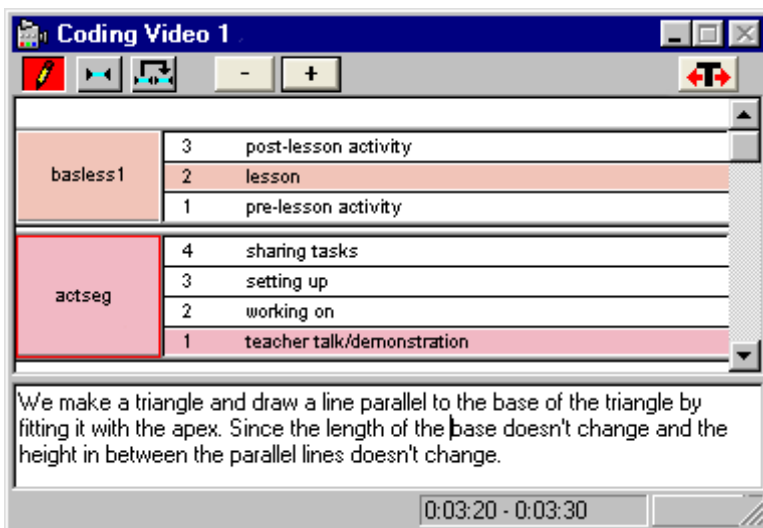
This window contains the instruments for coding and transcribing a video clip. It is made up of a combination of two horizontal panes. The upper pane is for coding, the lower one for entering and showing the transcription text. There is a movable divider between the two zones to regulate the relative height of the pane. One of the windows is always active. With the window title it is possible to recognize which one is. It is only possible to work in the active window. If you want to code, you have to activate the coding window; if you want to transcribe, the transcription window. You will find tips for activating windows later in the text.

In a newly opened project file the coding and transcription window is minimized on the lower edge of the work surface. It can be opened from there and brought into position.

When several clips are open, each one has its own timeline window, but there is only one coding and transcription window for all of the clips enabling you to switch back and forth among the clips.

A detailed description of the functions and handling of the coding and transcription window may be found [elsewhere](#).

•  
▶ View of the coding and transcription window with coding variables and transcript



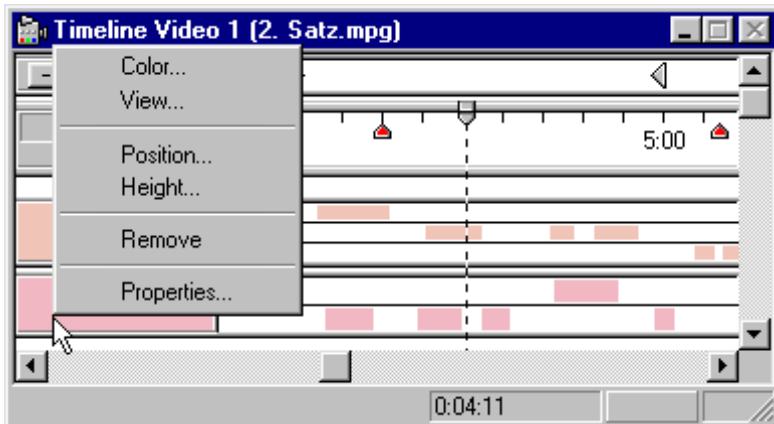
## ▶ Working with windows in Videograph

### shortcut pop-up menus

Some elements can be used when accessed by a shortcut pop-up menu and then selecting a command. The shortcut menu appears when the element is clicked on with the **right** mouse key and disappears when a command is called or the action is canceled. The elements which possess a shortcut menu are the **video window**, the **data bars** in the timeline window as well as **the coding bars** in the coding and transcription window.

- 

#### ▶ Example



Shortcut menu of a databar in the timeline window

## ▀ Customizing the work space

### Arranging windows

All windows in the workspace are built in the typical Windows-style and act accordingly. They can be moved in any direction and changed in size down to the minimum. They can be positioned next to each other, on top of each other or overlapping. It is, therefore, easy to put together a suitable window arrangement. All of the elements have a pre-determined position and size when they are produced for the first time, e.g. the timeline window stretches across the entire width of the work surface. The size can be halved without adverse effect should room be needed for another element.

All settings are stored in the project file, thus being permanently available.



## ► Customizing the work space

### Switching between windows

- Let's assume you are working with a project file that uses two video clips. You find the coding and transcription window, 2 timeline windows and 2 video windows on the videograph work surface. You want to start Video 1 and push the space bar and are surprised that Video 2 begins to play instead of Video 1. Or you push a cursor key to move the pointer of a timeline window and nothing happens. In such cases you forgot to activate the window you want. Windows can only receive key messages for active windows and only one of several windows can be active. If window A is active and window B is activated, then window A is made inactive. An active window can be recognized by the color of the title bar. In order to activate a window, just click on the window surface with the left mouse key, the best thing to do would be within the title bar. When working with the videograph you can also use special keyboard shortcuts (see below) to activate a certain window. In the Videograph program the change in the active window has another important effect: whenever the coding and transcription window is activated, thus deactivating a timeline window, or reverse, the available menu commands also change. In principle the following applies:
  - When you want to use functions with key or menu commands that are found in the timeline window, the timeline window must be active
  - When you want to use Videograph coding and transcription functions with key or menu commands, the coding and transcription window must be activated. This especially applies when:
    - you want to use the coding function the coding window must be active
    - you want to use the transcription function the transcription window must be active.
- In summary, activating the timeline window or the coding and transcription window has the following impact:
  - It is brought to the foreground should it be partially or completely covered by another window. Should it be minimized, it will be restored to its original size.
  - Its title bar takes on another color - usually blue
  - It receives an "keyboard focus" which means it receives all keyboard messages.
  - When switching between timeline window and the coding and transcription windows the menu commands change accordingly.
  - Should several clips be opened, each clip has its own timeline window, but there is only one coding and transcription window for all of the clips. You have to channel by switching over between the clips. This simply happens by switching the active timeline window. This selects the activated clip as the new goal for what happens in the coding and transcription window. The clip that is currently switched "on" is easily recognized: the clip whose name is in the title bar of the coding and transcription window is the activated one.
- When activating the video window the effects are ramified: It moves to the foreground, whereby the keyboard focus and is redirected to its timeline window as if it had been directly activated.

For functions that can be taken care of by mouse, it is not a problem since the mouse always affects the window it is in.

The following list shows which keyboard shortcuts can be used to activate the various windows. The TAB-key is probably the easiest to use. It is the fastest way to switch back and forth between the timeline window and the coding window.

## ► Customizing the work space

### Keyboard shortcuts for switching

<b>Keys</b>	<b>activated window</b>
Alt+1	Timeline window for Video 1
Alt+2	Timeline window for video 2 (Alt+3= Video 3 etc.)
Alt+V+1	Video window for Video 1
Alt+V+2	Video window for Video 2 (Alt+V+3=Video 3 etc.)
Alt+C	Coding and transcription window
Tab key ( <b>within timeline window</b> )	Activating the coding and transcription window
Tab key ( <b>within coding window</b> )	Activating the timeline window

## ► System requirement

### Media format

Should you have problems when playing back media clips (happens very rarely), certainly the correct codec for decoding the media format of the clip is missing.

Here are some hints for using media files when working with videograph.

Because Videograph no own codecs has integrated, the program uses the multimedia interface of Windows and is based entirely on the codecs that Windows provides. However, there are a number of video formats that require more powerful codecs (called "filters" under Windows) than those that Windows (also Windows 10) brings with it. They are available in large numbers; you just have to download and install them yourself from the Internet. There are comprehensive codec packages, which offer a suitable codec for almost every video format, and are still free of charge.

The best choice especially for videograph are the LAV filters, ready for download at following link:  
<https://github.com/Nevcairiel/LAVFilters/releases>

This tools are based on the famous FFmpeg library. The codec packet includes the "LAV Splitter" and "LAV Splitter Source" for demultiplexing files into their individual streams as well as the "LAV Audio decoder" and "LAV Video Decoder" for decoding multimedia files. Installing the LAV filters eliminates most playback problems. One of the reasons is that Videograph recognizes when the filters are installed and then sets up the video configuration so that Windows only uses the LAV filter instead of its own or other codecs when playing a video clip.

The filters are available as 32-bit and 64-bit versions. Since Videograph is a 32-bit program, you must always install the 32-bit version at the same time.

If you want to get an overview of other multimedia filters installed in the system, and perhaps also want to check one or the other for its suitability, read the chapter "Change filter merit"

But note: Videograph is not a DVD-Player, because the program is unable to read the specific structure of a video-DVD. Therefore are such media files, as typically being found on Video-DVDs and recognizable by the file ending \*.vob, generally unsuited for videograph.

Differently are Blue-ray disks containing high-resolution video material. Here the pure video files (normally with file ending \*.m2ts) can be severally and separately read-in and played back (given that a suitable H.264-codec is installed), although the coherent replay in whole will not function

## ▶ Loading and playing clips

### Main clip

A necessary component for working with videograph is a media clip. Before it has been opened the program is unable to function. When an existing project file is opened, videograph opens the appropriate clip automatically itself (see below for exception). When, however, you want to start videograph for the first time or when a new project file (\*.vdg) is to be opened, the first step is always the **loading of a video clip**. There are two equivalent menu commands for this: either the command "load clip" in the clip menu is used or the option "new" in the file menu is used. Then (in the usual file dialogue window in Windows) the path and the filename of the clip to be loaded are chosen.

The program will save pathname and the filename of the chosen clip in the new project file. Whenever you load the project file in the future, videograph will access the stored informations and load the clip itself. The pre-requisite for doing so is that the current path and the name of the clip to be opened are identical with the stored information. Should the clip have been moved to another path or the pathname have been changed in the meantime, videograph will not be able to find the clip. In that case at first the directory housing the project file is scanned. When videograph will not be able to find the clip here, it demands new information from you. Now you can also select a clip with another **filename** as hitherto, whereupon videograph is unable to determine whether the previous clip under a new name or a totally other clip is selected. Therefore before opening the clip the user is asked for confirmation, furthermore the clip must have at least the same length as stored in the project file, otherwise videograph refuses to open. Generally the filename of a clip should not be changed between project sessions to prevent unintended mix-ups in the clips.

To present the clip a video window is placed in the working space of the program and can be moved at will and varied in size.

The **volume** of the clip can be adjusted using slider controls located in a special toolbar. In order to make visible the toolbar, you have to select the command "fader" into the menu "view".

## ▶ Loading and playing clips

### Additional Clips

Additional clips may be integrated in a project file at any time. Their number is arbitrary, but due to the space required by the video window that is created and the hunger video files have for resources, the practical upper limit is three. Each clip has (as long as it is not coupled - see below for details) together with its timeline window its own unit and can be controlled, coded and transcribed independently from the other clips. In order to load an additional clip, choose the command **"load clip"** in the **clip menu**. Additional clips can be closed any time. To detach them from the project file, simply select the option **"close clip"** instead of the command "load clip".

The **volume** of the clips can be adjusted using slider controls. There are 5 sliders (1 for the main clip and 4 for additional clips) located in a special toolbar. In order to make visible the toolbar, you have to select the command "fader" into the menu "view".

## ► The video window

### View of the video window



<b>Play:</b>	Start the clip. Alternatively: Click the inside area of window with the left mouse key OR: Press spacebar.
<b>Pause</b>	Pause the clip Alternatively: Click the inside area of window with the left mouse key OR: Press spacebar
<b>Stop</b>	Stop the clip and puts it back to time position 0:00:00.
<b>Playback-Rate</b>	decrease or increase playback rate
<b>Mute</b>	Switch audio signal on or off
<b>Fit window</b>	fits window size to original aspect ratio of video.
<b>Original size</b>	restores original clip size
<b>Adjust position</b>	corrects the position of video window if it was moved accidentally outside of view.
<b>Frame accurate</b>	Used to coupling or adjusting clips in a frame-accurate manner.
<b>Properties</b>	Shows informations about the video clip, e.g. frame rate, media format, codec usw.
<b>Title bar</b>	Hold the border with the left mouse key pressed down and pull it to move the video.
<b>Border</b>	Hold the border with the left mouse key pressed down and pull it to change height or width of video clip.

## ▶ The video window

### Using the video window

The video window not only shows the current video clip, it also offers a shortcut menu with important operating functions. It also reacts to certain mouse movements. The shortcut menu is activated by clicking the right mouse key on any point inside the window. The same commands may also be found in the main menu under "clip".

A description of all the possible operations follows (or read about the hotspots in the previous section "view of video window"):

**Using the mouse**(always with the left mouse key pressed):

- Move the window:  
Take hold of the title bar and move the window to the desired position.
- Change the size of a clip:  
Move the left or the right border of the window to change the width; move the top or the bottom window border to change the height; pull on a corner of the window to change width and height at the same time.
- Alternately start and stop replay:  
Click inside of the video window or press SPACEBAR on the keyboard.

**In the shortcut menu:**

function	command
Start the clip	Play
Pause the clip	Pause
Stop the clip and reset him	Stop
Increase or decrease playback rate	Playback rate
Switch audio signal on/off	Mute
Fit window to original aspect ratio	fit window to image (1:1)
Restore the native clip size	original image size
Correct misplaced video window	Adjust window position
Linking or adjusting clips frame-accurate	Frame-by-frame alignment/Frame accurate adjustment
Informations about the clip	Properties

The **volume** of the clip can be adjusted using slider controls located in a special toolbar. In order to make visible the toolbar, you have to select the command "fader" into the menu "view".

In the main menu of the clip you will find also the option "**Media format**". This command can be used to determine the unit ( frames or milliseconds) when linking clips together and aligning both time-accurate ( see [here](#)).

## ► Opening and playing of video clips

### Coupling clips

Videograph offers the option to play two or three clips at the same time. One clip is the "Master clip" that controls the other clips ("slaves"). **To calibrate**, the pointer in the time ruler of the clip to be coupled is set at a certain time. The **difference** between the value of the master clip and the clip to be coupled shows the offset between the two.

When, e.g., the master clip X is at position 0:00:00 and the slave clip Y is at 0:00:10 it means: whenever master clip starts or is at position  $x_1$ , then slave clip starts or is at position  $y_1 = x_1 + 10$  seconds; it is calibrated by setting the master clip X at 0:00:10 and the slave clip Y at 0:00:00, the starting time for clip Y is always 10 seconds before the starting position for clip X ( $y_1 = x_1 - 10$  seconds). The master clip is always the "Video 1" video, i.e. the clip that is opened first when generating a project file. It follows that there is only one master clip, but there can be several slave clips and that the video clip functioning as "Video1" can never be used as the slave clip.

**In order to couple a clip**, you have to activate its timeline window and select "couple" from the "clip" menu. Coupled clips can never be replayed independently of each other. They start and stop at the same time. Every movement that is made with the master clip (forward, backward, altering the time position, stop the replay, start the replay) also has a direct effect on the slave clip, the synchronization, also during replay, is always exact down to the second. Whether a clip has been configured as a slave clip can be seen in its timeline window: the buttons of the time ruler have been faded out. Furthermore when coupled clips are paused, the indication of their position in the status bar of the timeline window is accurate to a millisecond ( in contrast to uncoupled clips). The indication for the slave-clip additionally shows the time lag to its setpoint, i.e. it shows its deviation from displacement value that you have adjusted in the dialogue.

A slave clip exclusively serves as an additional source of information when viewing or analyzing the master clip. It can neither be coded nor transcribed. No playback intervals can be chosen. In order to **rescind a coupling**, activate the timeline window of the clip and select the "uncouple" command in the "clip" menu.



## ► Opening & playing of media clips

### Linking clips frame-accurate

Clips that are to be played linked together can be **aligned frame-accurate** before coupling when necessary, so that it is possible to listen to both without a disturbing time delay as is, e.g., the case when filming with 2 cameras that did not start at exactly the same time. On clips that are not composed of frames like e.g. **audio clips, milliseconds** instead of frames are applied as measure on adjusting. Optionally you can determine the unit of adjustment in advance by means of the command "Media format" from the "clip" menu.

#### How to proceed:

- Select the command "frame-accurate adjustment" in the timeline window of the clip acting as a slave (characterized as Video 2, 3, etc.) in the "clip" menu. This command can also be found in the shortcut menu of the video window. It is recommended that the playback interval be deactivated as a trial run (see below) is not possible otherwise.

Note: The command may also be selected while Clip 1 is activated, but only in order to align the picture frame-accurate and not to couple them. In this case the clips may not be coupled before.

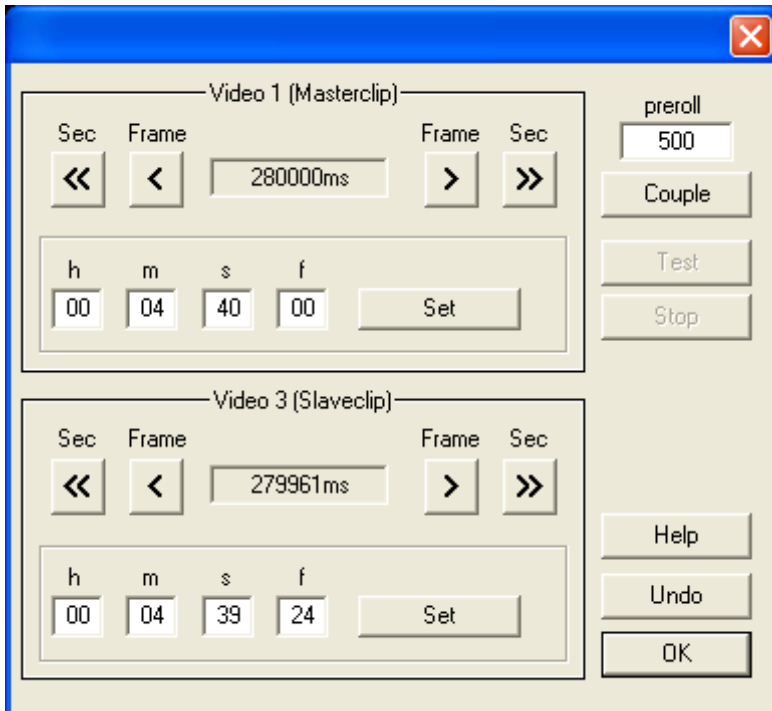
- A dialogue window appears and in the upper half there are the controls to position the pictures of the master clip (video 1) frame-accurate; in the lower half there are the controls for the slave clip (see below for a more detailed description).
- After the time positions have been set and synchronized for both clips, click the COUPLE button to link clip 2 (slave clip) to clip 1 (master clip).
- You can now test how perfect the synchronization is. A click on the TEST button starts replaying the linked clips so you can test how exact the time positions are and correct them if necessary.
- Press STOP to end the trial run. Both clips will rewind to the starting position. If you are satisfied with the result, press COUPLE again and then OK. If you are not satisfied, readjust and repeat the test.
- Before closing the dialog box, make a note of the set time positions. You can then disconnect the clips and reconnect them without having to search for the correct setting.
- In order to cancel all operations, press the UNDO button. This resets the clip positions to their original position when the dialog window was opened.
- Depending on the system environment there can be a slight delay between starting the replay of the master clip and of the slave clip after coupling them in spite of exact adjustment. By entering a pre-roll time of  $> 0$  or by increasing the value this effect may be overcome. The pre-roll default value is 0 milliseconds, the maximum value 2000. Should the synchronization not be satisfactory despite frame-accurate adjustment, experiment with the pre-roll value.

What effect does the pre-roll value have? It delays the replay start by the set number of milliseconds which allows the system more time to access the slave clip. With a high pre-roll value, e.g. 1500, there will be a noticeable break between the play command (e.g. by pressing the space key) and the actual begin of the replay.

Which factors can make an increase in the pre-roll time necessary? When the clip is on a CD it can require a greater pre-roll time than when it is on the hard disk. When both clips are on a CD the necessary pre-roll time depends on the differing access times for the drives. The access times may, however, depend on the position of the clip, i.e. where the sector physically is on the CD that has to be read. It is, thus, possible to even have differing pre-roll values within the same clip. All of these problems can be avoided to a great extent when both clips are on the hard drive.

- Various parasitic inductions can cause the quality of the synchronization to deteriorate during replay. A short pause in the replay, e.g. by pressing SPACEBAR, suffices to re-synchronize the clips.
- Please note: Should the clips already be coupled when the dialogue starts (can be seen at that the button for coupling, see figure at bottom, is labelled with DECOUPLE instead of COUPLE and functions too in that manner), also the time positions on adjusting are coupled. In that case you have to uncouple the clips in order to move the clips independent of each other.
- In order to couple multiple slave clips, you have to open the dialogue for each further clip again. I.e. open the dialogue, adjust and couple the first slave clip, then close the dialogue, open it again - without decoupling clip 1 - for adjusting and coupling the second slave clip, then close it etc. The more slave clips are to be coupled, the more advisable is it to make a note of the set time positions, so that you have not to retest the correct setting when by mistake a clip is decoupled.
- When coupled clips are paused, the indication of their position in the status bar of the timeline window is accurate to a millisecond (in contrast to uncoupled clips). The indication for the slave-clip additionally shows the time lag to its setpoint, i.e. it shows its deviation from displacement value that you have adjusted in the dialogue.

Here you see the dialog window to couple two video clips frame-accurate.



Moves clip backward 1 second



Moves clip backward 1 frame.



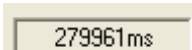
specifies current position of clip in milliseconds



Moves clip 1orward 1 frame.



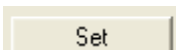
Moves clip forward 1 second.



specifies difference between position of clip 2 (slaveclip) and clip 1 (master clip) in milliseconds



Here you can enter a frame-accurate position (h:mm:ss:ff) for clip 1 (masterclip) as numerical value. In order to set the clip to the entered position the button "Set" must be pressed after that.



Moves clip to the entered position.



After the time positions have been set and synchronized for both clips, click here to link clip 2 (slave clip) to clip 1 (master clip). Thereafter you can test how perfect the synchronization is.

Test

starts replaying the **coupled** clips so you can test how exact the time positions are and correct them if necessary. The procedure only functions when the **playback interval is deactivated**.

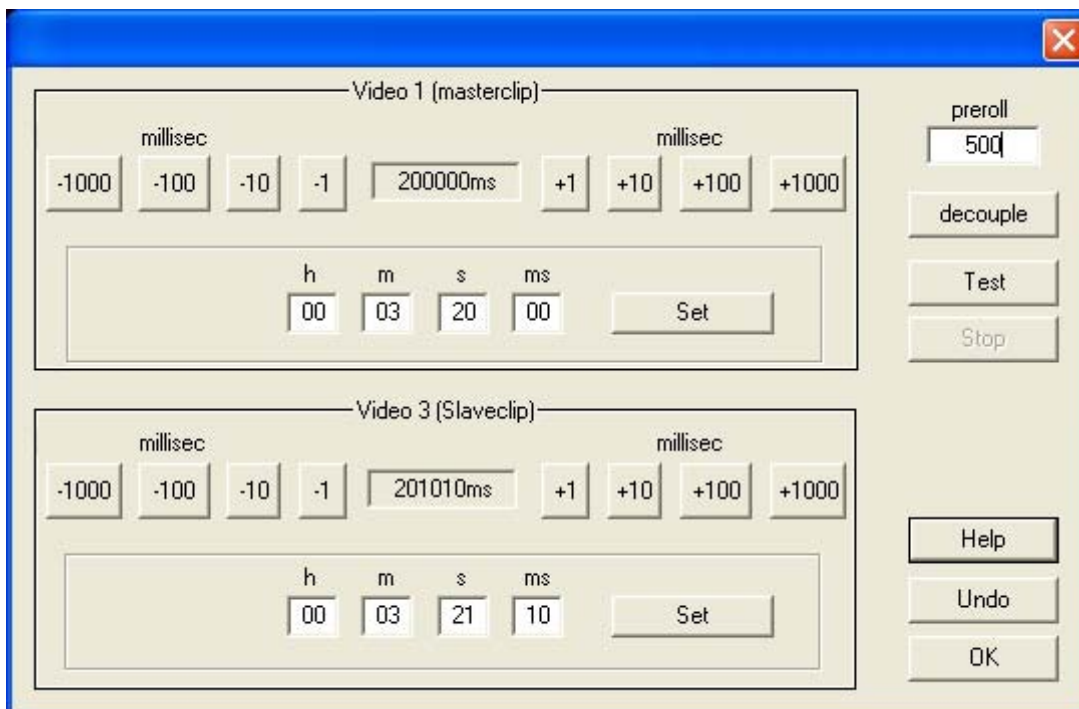
Stop

Press here to end the trial run. Both clips will rewind to the starting position. If you are satisfied with the result, **press COUPLE again** and then OK. If you are not satisfied, **readjust** and repeat the test.

Undo

Press here to reset the clip positions to their original position on opening the dialog dialogue window.

Should frame-accurate coupling be impossible, because e.g. one of the clips is a audioclip, instead of the preceding window the following dialogue is presented. Its functions exactly as the dialogue for frame-accurate coupling, but instead of frames the measure for adjusting are milliseconds. This dialogue also appears if you have determined milliseconds as unit in advance via the the command "Media format" from the "clip" menu



## ► View of the coding and transcription window

Scrollbar: enables the user to exchange the position of coding bars in a ring-shaped manner.

Coding window

The splitter bar separates the coding window from the transcription window. The user can drag him with the mouse to change the relative sizes of the two panes.

Transcription window

Indicates the interval to which the showed transcript text belongs.



Important protection against unintentional coding and transcribing. In order to make possible each action, the button must be pressed down, making a red background visible.



The function of this button is dependent from modus. During transcribing it serves as "Fix-Button" to save entered text, during coding it provides for an automatic displacement of the playback interval to the next start position.



Provides in case of time-sampling for the automatic repeat of last coding in the following playback interval. Applies only to coding variables that were selected for it.



Displaces the coding interval backwards one length.



Displaces the coding interval forward one length.



This button serves to code using variable time intervals ( so-called turns). Generates In-Point and Out-Point of a turn.

basless1	3	post-lesson
	2	lesson
	1	pre-lesson

Every variable is represented by a coding bar. It consists of a head and category fields with the variable values and value labels. Coding is done with a click of the mouse in a category field. When you click with the right key of the mouse in the bar head you see a shortcut menu with options with which the characteristics of the bar may be altered. The positions of the individual bars can be exchanged among each other at will.

actseg	4	sharing tasks
	3	setting up
	2	working on
	1	teacher talk/demonstration

This coding bar is the active one. Its left part has a red frame. An active coding bar signals which variable is being coded. You can activate a coding bar by left-clicking on his heading section during pressing the SHIFT key, or when coding (write protection must be switched-off) by clicking on a category field.



The titlebar fullfills two tasks: It indicates the clip determined for coding and transcribing (important only when several clips are opened) and it reveals whether the coding window or transcription window is activated.

## ▶ The coding and transcription window: an overview

This window contains the devices for coding and transcribing a video clip. It is made up of 2 horizontal panes. The upper window is for coding, the lower to enter and show the transcription. Between the two zones there is a movable divider. The relative height of the pane can be regulated (by mouse). One of the windows is always active. To know which one it is, you must check the window title. It is only possible to work within the active window - to code activate the code window, to transcribe the transcription window. To find out how to activate the windows, see [here](#) .

In a newly opened project file, the coding and transcribing window is minimized on the lower edge of the working surface. From here it can be opened and brought into position.

When several clips are opened, each clip has its own timeline window, but there is only one joint coding and transcription window which means having to switch between clips. This is done simply by activating the timeline window or the video window of the clip that is to be worked on in the coding or transcription window. The "activated" clip is easily recognized: the clip whose name appears in the title line of the coding and transcription window is the one being worked on.

Another characteristic of the window is that it has its own menu. The same is true for the timeline window. Only the menu of the activated window is visible and accessible. Should you search for a command in vain, check to see which modus is open, either clip modus (= activated timeline window) or coding/transcribing modus.

## ► Using the coding window

### Defining coding variables

The coding process begins with defining the variables. The following elements have to be declared:

- **Names of the variables.** In order to guarantee a minimum of compatibility with the SPSS syntax, Videograph only allows names with a maximum of 8 characters. Following all of the other rules prescribed by SPSS is the user's responsibility.
- **Amount of scale values (categories or levels).** The allowed area in Videograph ranges from 1 to 40.
- **Range of values.** As only numerical coding is allowed, every category or scale level has to be represented by a whole numbered positive value. Other than the minimum value is 0 and the maximum 99 there are no rules for choosing the values as the theoretical measuring interpretation of the scale (nominal scale versus ordinal scale) is the responsibility of the user.
- **Value Labels.** A label (verbal description) has to be given for each scale value. The label can be 255 characters long.
- **Variable Label.** What is meant is a verbal description of what the variable is supposed to measure. This label can also be 255 characters long. The length should, however, be chosen to allow the data to be exported to SPSS (should this be planned). Awarding variable labels is **optional** but recommended.

In order to define a new coding variable, go to "coding" menu and select "define new coding-variable". The rest of the procedure is self-explanatory.

Following the definition, the variable is inserted in the coding window as a so-called **coding bar**. At the same time in the timeline window a connected **data object** will be created, a so-called **data bar**, which stores and graphically shows the data created by the coding.

There can be any desired number of data bars for each coding bar and thus for every coding variable. Practically this means that any number of individual coding passes can be carried out for one variable without additional effort, using, for example, various judges, and still being able to keep the data together in one file. As every data bar gives a clear graphic representation of the data, it is also possible to make simple visual comparisons, e.g. to estimate the judgement concordance. A new data bar can be created ad hoc at any time. The necessary command can be found in the shortcut menu of the coding bar.

When exporting data to SPSS every data bar is treated as its own coding variable so that (when exported entirely) the number of coding variables in the SPSS file equals the number of coding variables in the videograph times the number of connected data bars.

All of the defining elements with the exception of the number of categories can be quickly and easily modified at any time. The necessary command is in the shortcut menu of the coding bar.

**Another method** to define variables is to **import them from another project file**. This option is especially helpful when the same variable set is to be used for a number of clips. To import data, select in the "coding" menu the command "import coding-variables". You then have all of the file variables you want to import to select from. Should a variable number already be in the project file, you first have to rename the variable before you can import it.

Note: From an SPSS file only **data** (bars), but no variable **definitions** can be imported. The variable, for which the data is to be imported, has to already be defined. The reason is that an SPSS file cannot contain any clear information about the allowed value range of variables.

## ► Using the coding window: duplicate the coding variables

Another way to create new coding variables is to completely or in part **copy the defining attributes of a variable that already exists**. In the shortcut menu of every coding bar there is the command "open as template...". This leads to a dialogue box which lists all of the definition characteristics of the variable from names to value range. Adaptions can be made here or things can be left as is. All that is necessary is to give a new name to the variable. After doing so, just push OK-Button and an new variable is built.

Where several "individual units", e.g. students, are to be coded with the same categories but where a separate coding bar is to be available for every unit, a lot of typing can be spared with this option.

## ▀ Using the coding window: Importing coding variables

**A further method** of defining variables is to **import them from another project file**. This option is especially useful when the same set of variables is to be used for a number of clips. To import select "import coding-variable " order from the "coding"" menu. You then have all of the variables from the file you want to import available for selection. Should the name of a variable already be present in the project file you have to first rename the variable before you can import it.

Note: From an SPSS file you can only **import data**(bars), but not import **definitions** of variables. The variable for which the data is to be imported must already be defined. The reason is: an SPSS file contains no clear information about the allowed value range of a variable.



## ► Using the coding window: Importing coding variables II

In larger studies one usually has more video material to analyze. Several project data files that are built up in the same way are necessary, especially when the configuration of the coding variables is concerned. In the option menu of the timeline window there is the "**transmit project settings**" command. The current project file can be used as the pattern to quickly create new project files to analyze further videos that are to be coded with the same instrument as the current clip.

Procedure:

1. Create a project file and define the required coding variables using the point in the menu "define new coding variable" from the coding window.
2. According to the available possible variations configurate the desired looks of the coding bars (color, arrangement, sequence, etc.) and the data bars in the timeline window.
3. Access the command "transmit project settings".
4. Select from the file dialog window the clip for which the new project file is to be created.  
Should you not close the dialog window with OK, but with ABORT then the currently opened clip is the chosen one.
5. When a second clip is opened the dialog window appears for the second time so you can select a further clip.  
Should you not close a dialog window with OK, but with ABORT then the currently opened clip is the chosen one.
6. The last step is to assign a name for the project file to be created and to select a directory to store it. When you close the file dialog with OK, Videograph creates a "fresh" new project file with the same coding variables and with identical and identically arranged coding and data bars. It makes no difference whether the project file that serves as the pattern already contains data or transcripts as these are not copied into the new file.

## ► Using the coding window: Codingbars and databars

Every coding variable is represented by a coding bar. It consists of a head and, in accordance with the variable values, category fields one upon the other. Coding is done with a click of the mouse in a category field.

The head of the column contains the shortcut menu which allows access to a number of options with which the characteristics of the bar may be altered. It also forms a "handle" which the mouse may grip to move the columns to another position. The name of the variable is in the heading. It can be faded out (using the shortcut menu) which diminishes the width of the head leaving more room for showing the labels in the category fields.

- The coding bars can either be active or inactive. A red border around the head of the bar shows it is active. Activating or deactivating a coding bar activates or deactivates its data bar in the timeline window. An inactive coding bar can be moved and adjusted by means of the shortcut menu, but it allows no data transfer and, thus, cannot be coded. An active bar becomes passive as soon as another bar becomes active. A coding bar is activated or deactivated by pushing the SHIFT key and clicking the bar head with the left key of the mouse. When **coding** you do not worry about it. When you click in a category field of a bar it is **automatically** activated.
- When you click with the right key of the mouse in the bar head you see the **shortcut** menu with options: you can change color, position and height of the bar. Also the commands to create new data bars, to delete a coding variable or to modify the variable definition are found in the shortcut menu.
- To have more room to show the category labels there are commands to hide both the name of the variable in the head as well as the numerical value in the category fields. The variable label can be made visible or invisible. Should only the appearance of an individual bar need to be changed, the necessary order can be found in the shortcut menu of the corresponding bar. Should the change need to affect all of the bars at the same time, then choose the global command "View of coding bars" in the option menu of the coding window.
- Coding bars can both be vertically and horizontally **relocatable** and - in their sequence **commutable**. A bar can be relocated by "grabbing" its head with the left mouse key, holding the key down and moving the head to the desired position.

A coding bar can cover an entire window in width (single bar grouping) or it can be paired with the neighboring column on the right (double bar grouping). There should, therefore, be a satisfactory solution for the amount of required space: All of the coding bars can be stacked one on top of the other, or all of them with double columns, or a mixed format can be chosen. If all the coding bars still do not all fit in the visible area of the window, the scroll may be used (see below).

The positions of the individual bars can be relocated among each other at will. A bar A immediately moves to the position of another bar B when its heading is pulled over the B heading and "dropped". In order pair a bar A with another bar B on the right, pull the heading from B over to the right half of A. To convert a right-handed bar into a stand-alone bar, press and hold the **CTRL** key and pull it across its left neighbor.

A right-handed bar should not be higher than its neighbor on the left as it would then cover part(s) of the following bar or be covered by it. Should this occur, simply exchange the positions of the left and right bars.

There are a few differences in how to relocate bars depending on whether they are left-handed or right-handed one and whether they are to be moved in a vertical, horizontal or diagonal direction. These differences can easily be discovered on your own.

It is generally true that **vertical** relocation of left-handed or stand-alone bars always offers two alternatives:

Keeping the CTRL key pressed when relocating results in a dual exchange: Bar A moves to the bar B position and vice versa. The position of all of the other bars including the right-handed ones remains unchanged. Should the CTRL key not be pressed, a domino effect results, whereby the right-handed bars remain attached to their left neighbor and move with it.

When horizontal or diagonal moves are made and when moving right-handed bars, the result is always a dual exchange.

The best would be to try out the various possibilities.

**Note:** Other ways to expand the available space are to fade out the name in the bar heading and to suppress indication of the numerical values in the category fields.

There are other methods to exchange the positions of the coding bars:

- When there are more coding bars than can be shown in the window at the same time, then the vertical **scrollbar** in the coding window is used, whereby the coding bars change their position in a circular motion. They can not only be scrolled stepwise by moving the scroll thumb up or down, but also moved **immediately** into the visible area by aiming at its position in the scrollbar and clicking there.
- In the shortcut menu of the coding bar select the order "position"
- A third available method is using the **TAB** key. By it the coding bars can be moved up and down (when

the **SHIFT** key is pressed) like they were on an assembly line one by one. Pivot of the pass is always the **active** coding bar, i.e. one is free to determine which coding bar is to be included in the run through and which is not. Caution: The method functions **only** when coding, i.e. the writing protection must **rescinded** and a coding bar must **be activated**. Outside the coding procedure the tab has a completely different function: it activates the timeline window.

The **databars** in the timeline window have the same characteristics the coding columns have. They can be relocated by pulling with the mouse, by using the scrollbar or by pressing the tab key (only while coding) in any desired vertical position or sequence. The only important difference lies in the nonexisting possibility of a dual column layout. In other respects the rules for handling are almost 100% identical. Another slight difference: With a click on the right in the data bar head when the SHIF T key is pressed the variable label is made visible.

Another very practical method to **control** the databars is available in the "options" menu of the coding window under the entry "databar navigation". The timeline window then automatically sets the data bar in **first place** that belongs to **the active coding columns**. This makes it possible to see the coding data of prent interest without having to intervene in the timeline window even when coding. You can comfortably page through the coding data from the coding windo w by simply clicking one coding bar after the other with the left key in the head (or with the tab key). In order to change the direction, keep the SHIFT- key pressed.

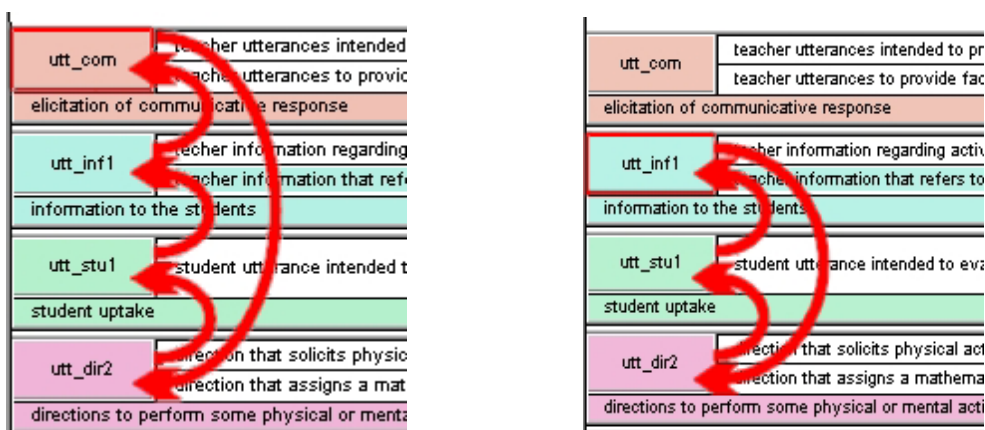
If there are several **data** bars for one coding variable, then the bar that is supposed to receive the data must once **manually** be activated. To do so, press the **SHIFT**key and click (like for the coding bars) with the left mouse key in the bar heading.

Example: Effect of **TAB** key whilecoding beim Codieren

utt_com	teacher utterances intended to provide teacher utterances to provide elicitation of communicative response
utt_inf1	teacher information regarding teacher information that refers to information to the students
utt_stu1	student utterance intended to elicit student uptake
utt_dir2	direction that solicits physical action direction that assigns a mathematical directions to perform some physical or mental action

utt_com	teacher utterances intended to provide teacher utterances to provide elicitation of communicative response
utt_inf1	teacher information regarding active teacher information that refers to information to the students
utt_stu1	student utterance intended to elicit student uptake
utt_dir2	direction that solicits physical action direction that assigns a mathematical directions to perform some physical or mental action

:



## ▶ Printing coding data

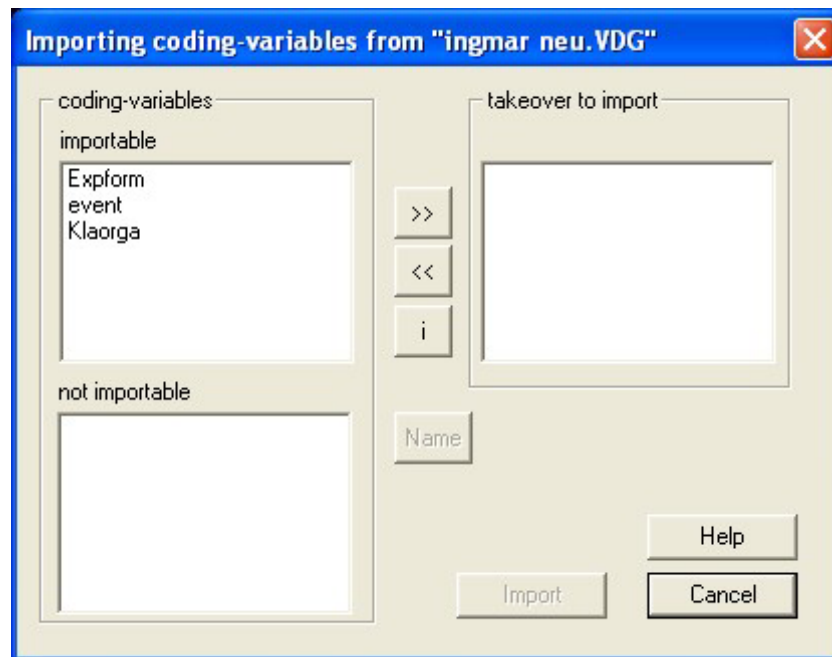
A coding variable in the coding window does not store data, rather it is an interface that accepts input, classifies it and then sends it as data to the databar of the timeline window. This creates a graphic presentation of the transmitted data in the form of a time diagram. All data bars in the timeline window can be printed.

To get a print, activate the timeline window and select the "print databars" command in the file menu. Using the command "print preview" you get a preview of the printed result. With the "print set up" command you can, you can set the page length and width from "portrait" to "landscape" in order to enlarge the scale of the image.

In order to be flexible in the choice of the databars or the time period to be printed, all available data bars are not categorically printed, only the **visible** ones, whereby the visibility of the bars depends on the height of the timeline window and on its layout. Or stated differently: the data bars that are not to be printed must **completely** disappear from the visible area of the timeline window. This is not difficult to do as the databars can be resorted in any way. The data bars to be printed are relocated in the front positions and/or the timeline window is moved to a height in which only the data bars that are to be printed are visible. Should the maximum height of the timeline window not suffice to make all of the desired elements visible, then two or more printouts are used.

The same is true for the horizontal axis. Only the data in the **time section** visible in the timeline window is printed. It is then possible to vary the timespan at will, either by scrolling horizontally or by widening the timeline window. The same effect may be reached by altering the time increment. The smaller the increment is, the longer the time period shown is. When you want to print the entire stretch of time at once, you can set the time period to "all". Yet, in this case, the legibility of the printout mainly depends on the length of the clip, the width of the timeline window in Pixel (i.e. the screen resolution), the length of the coding interval, and - last but not least - on the width of the paper. Should the printed data points be too small, then the entire time period should be divided into two or more segments. These should be printed individually and then pasted together. The time period window shown in the timeline window is usually scaled when printed so that it covers the entire (printable) width of the paper. The wider the paper, the wider the image.

## ► Help on import dialogue



The **upper left window** contains the list of the **importable** variables, in the **right window** there are the names of the variables that are to be **imported**. The lower left window shows variables that cannot be imported because they do not have a suitable format. These are usually variables whose name has already been used. In this case you can rename the variables.

**Using** the mouse to select the variables that you want to import from the **left** list.



Click here to move the selected variables from the **left** to the **right** window.

You can also **reverse** the procedure. From the **right** window you can select the variable you want to **delete** from the import list.



Click here to move the selected variables from the **right** window to the **left** one.



A variable that cannot be imported can usually be made importable by renaming it. Select the variable and click here to rename it.



Select a variable and click here to have a description of the variable.

## ► Using the coding window: The coding procedure

### Fundamentals

In order to be able to code, the coding window must be activated and the writing protector rescinded, i.e. the writing protector button must be pressed down, making a red background visible.

The coding procedure as such is simple: Select a category in a coding bar and click in the category field with the left mouse key or - using the keyboard - enter the numeric value shown in the category field. Coding with the keyboard, however, only functions with one-digit values.

While the coding is in progress, the chosen category field is accentuated in color to signal its status. The coding is completed when the clip has reached the end of the playback interval (when **time sampling**) or when the same category field has been clicked on a second time (when **event sampling**). In both cases the category field returns to its original state. Or the coding is completed by clicking on another field to change to another category.

The videograph treats the variable categories as being **disjunctive**. Within a variable it is not possible to code several categories at the same time. Categories where a conjunct incidence may be expected have to be packed in various variables.

When trying to code the error message "destination bar not marked" might appear. This happens when a databar has not been chosen as receiver for coding data. This effect occurs when several databars are established for the same coding variable and videograph is unable to decide on its own which column is to receive the coding data. In this case, keep the SHIFT key pressed and click on the head of a data bar to activate it.

Every coding is immediately graphically shown in the data column. This provides continual control as well as visual feedback for one's own actions. On completion of the coding (or at any time in between) the data may be seen in a longitudinal overview by simply running along the time ruler in the timeline window.

Normally a category field in a coding bar is accentuated in color only when it is just coded. However the coding bars can be used as well to show all existing coding data. When activating the option "Show data" in the options menu of the coding window it follows that this category accentuated in the databar is every time also accentuated in the coding bar.

## ► Using the coding window: The coding procedure

### Time-Sampling

Time-sampling exists when a playback interval is **activated**. Fixed units of time that are prescribed by the playback interval are used for coding. A coded value always applies to the entire interval whatever interval second is used for conducting the coding. A graphic indicator is the length of the colored stripe which is created with every coding in the data bar of the timeline window. Every colored stripe is just as long as the playback interval it is based on.

This fact has some technical advantages. It is possible to recode within the playback interval at any time without having to rewind the film to the beginning of the interval. The same is true when a coding is to be deleted. The cancel procedure also always affects the entire interval. Furthermore, it is never necessary to reset the coding by hand (as in event sampling) as this is done automatically at the end of the interval (for exceptions see below)

In Videograph coding and deleting (canceling) are like a switch. A coding is deleted or relocated by **repeating**, either by entering the same value via keyboard or by mouse clicking on the same category field of the coding bar as when coding.

As with the function of the playback interval the clip plays to the end of the interval and then stops. Using the key combination CTRL+CURSOR RIGHT or CTRL+CURSOR LEFT the playback interval can then be moved to the next position. It is also possible to use the +or -buttons ("shift playback interval") in the toolbar on the upper edge of the window (or in the timeline window) (see the [view](#) of the coding window).

Whoever finds the interim stop to be a hindrance can press the "**no play stop**" button. It provides - as soon as the playback interval is runned trough - for his automatic displacement to the next start position without pausing the clip (see view of the [coding window](#)).

Coding can take place in a play loop when necessary. Using the "**loop**" button in the timeline window it is possible to repeat an interval in an endless loop until it is stopped.

"**Lock playback area**" button in the timeline window is also potentially important. Should you wish to learn more about this, read [here](#). In this connection remember that depending on the length of the video at the end an remaining interval may result which can cause a shortening of the coding - and thus the playback interval. You should also not forget to reset the playback interval to its original size before continuing with the coding.

Normally a coding is only valid until the end of the playback interval in which it takes place. At the end of the interval or when switching to the next playback interval it is erased. This can be avoided by pressing the "**repeat coding**" button in the coding window (see the view of the [coding window](#)). When the button has been pressed and coding follows then it will not be erased when the playback interval is moved, rather it remains until the user re moves it. This is a **selective**function and only effects variables that are explicitly released for it. Furthermore, it only works when - without pausing the clip at the end of the interval - there is a fluid change from one interval to the next. This can best be done by either activating the loop modus or by switching on the automatic interval displacement (see above). There may also be no coding for the time between the current playback interval and the end of the clip. This rule is to avoid accidentally over-writing existing coding.

As mentioned earlier, which coding variables are to be used or not for the automatic repeat function can be **selected**. In order to allocate (or to block) one or more coding variables for repeat coding, select the "**repeat codings...**" command under "options".

Time sampling is not only possible during replay. It can also be carried out when the clip is paused.

## ► Using the coding window: The coding procedure

### Coding on the basis of transcription intervals

A special kind of time sampling is possible when there is a transcript. The transcription intervals can then also be used as time units for coding. When the transcript is divided into equally large time intervals then the procedure is trivial: just switch on the playback interval in such a way that it is just a large are the transcription interval.

When, however, no set times units were used in transcribing, but rather the intervals were formed according to other criteria, then the sections to be coded usually vary in length so it is not possible to use a set interval length.

Videograph offers a control mode that functions just like coding with a constant time interval whereby the playback interval automatically adjusts its length to the available transcription interval.

#### How you have to proceed

- Set the time unit so that a playback interval that is at least as long as the shortest transcription interval can be shown. Remember: the shortest length you can set a playback interval for is the time between two tick marks on the time ruler. When the time unit is 1 minute e.g. then playback intervals shorter than 1 minute cannot be set. When during coding you try to get to transcription intervals that are shorter than the shortest possible playback interval, then you get an error report.
- Activate the playback interval. Its position and length are unimportant at this moment, but it has to be activated.
- Choose the "use transcript intervals" command from the options menu in the coding window. The playback interval is then automatically placed where the first transcription interval begins and is lengthened or shortened so that it exactly matches the transcript interval.
- Now you can begin to code. The same rules apply as for coding with a constant coding interval.
- To change to the next or to the previous transcription interval, use the buttons with + and with - in the coding window as usual. This means that the playback interval and the transcription interval are always exactly the same. Careful: do not use the buttons in the timeline window as these maintain their usual function, i.e. to move the playback interval forward or backward by 1 length.
- When the transcript is still incomplete or it has gaps, then there are untranscribed time intervals at the beginning, the end or elsewhere. Videograph then does the following:  
When there are no transcribed intervals after the playback interval, then it (and the clip) is moved forward 1 length, the size does not change. This corresponds to the coding with time intervals of the same size. The same rule applies in the other direction.  
When the gap is between two transcribed intervals then the playback interval is set up just as were the case with a transcribed interval.



## ► Using the coding window: The coding procedure

### Event-sampling

When there is **no** activated playback interval, the length of the validity of a coding is variable. It begins with the second the coding starts and is extended second by second as the clip is running and ends when the user switches off or changes the coding, pushes the turn-button or carries out an action which moves the position marker in the time ruler to another position. At the same time the colored strip in the data bar of the timeline window which graphically demonstrates the coding process permanently grows as time passes. The coding can be set or reset when the clip is running or when it is paused, but coding intervals longer than 1 second can only be created when the clip is running (in contrast to time sampling). The difference to time sampling is that every coding not only has to be switched on, it also has to be switched **off**. Coding is begun with a click in a category field (or by entering a scale value with a key) and is ended when the event currently coded is over with a click in the same or another category field; in the second case the new coding simply replaces the old one without having to switch it off as the categories of a coding variable are all disjunct.

There are two ways to delete or overwrite already completed coding:

- Coding is repeated by replaying the period to be recoded from beginning to end. Codings can only be overwritten with this method, but not erased.
- The +/--buttons or one of the equivalent keys PAGE UP/PAGE DOWN is used. With them it is possible to "scroll" through the coding intervals (see [here](#) for more details), i.e. you can jump from the end of the interval to the beginning and vice versa as well as from one interval to the beginning of the previous one or to the following one. The left border marker of the playback interval is set at the beginning and the right one at the end of the headed coding interval as a special "service". After a double click in the playback interval to activate it, it is possible to easily and quickly carry out recoding without having to replay the entire coding interval from beginning to end. It is possible to overwrite as well as to delete codings with this method. The length of the interval can also be modified.

Every time you switch off a coding - by mistake or on purpose - the coding interval is also closed. Every when the coding is continued with the same value a new coding interval is created. Depending on the procedure it is also possible that several intervals with the **same** coding value are next to each other. This is not visible in the data bar. When this bothers you or it is undesired for data export, you can "combine" these segments in a single coding interval:

- In order to see whether and where such interval sequences occur, select "Special" and the "mark interval boundaries" command there in the option menu of the coding window. This shows the borders of the coding interval in the data bars are marked red.
- Under "special" you will also find the "combine intervals" command. There are 3 possibilities: "**local**", "**global**" and "**select...**". "Local" only combines intervals in the active data bar which are around the momentary clip position. The "global" command combines all of the intervals in the active data bar next to each other and with the "select..." command you can select **several** data bars as the goal. Please note: of course these commands only function without write protection and when the playback interval is **deactivated**.

When coding 2 or more variables parallel it will not be possible to set the begin or the end of the coding exactly to the second. In this case you can use the "fill gap" command in the option menu under "special". This command removes a gap that is present between the current coding ( e.g. coding on the current clip position) and the end of previous coding interval. The command only affects the active data bar and the position pointer must be within the coding interval to be lengthened.

A potential disadvantage of the coding procedure described here is the following: when there is more than one coding variable and these variables are coded completely independent of each other, then almost every variable necessarily has other coding intervals. When exporting data with the "parallelize intervals" option intervals created are partitioned in such a manner that the arising transformed intervals are exactly the same between all exported coding variables. The temporal information about a coded category is, however, divided among several data rows which makes the data matrix confusing and certain calculations difficult. When "turns" are used as the coding unit, this effect is avoidable.

## ► Using the coding window: The coding procedure

### In-points and out-points (turns)

#### How to proceed

The **entire** area between one "in-point" and an "out-point" is always coded. An **in-point** is set to define the beginning of a new, still **open**-ended coding interval, i.e. either at the beginning of the clip or whenever it is time for a change in coding. An **out-point** at the same time closes and codes the interval.

In order to avoid gaps in coding, **every out-point** in Videograph is also a **new in-point** so that the coding intervals are automatically continuous when desired. When not, any other in-point may be selected.

#### How is an in point or an out point set?

They can be set both when the clip is running and when it is paused. The only pre-requisite is that the playback interval is **deactivated** and the write permission is switched **on**.

The pivotal point is the **"turn"** button which can be found rightmost in the toolbar of the coding window. Press **1x** sets an **out-point**; press **1x** while pressing the **CTRL** key sets an **in-point**. The alternative is to press the letter **"T"** on the keyboard. In-points can be set anywhere and everywhere. Out-points are only possible when there is already an in-point which precedes (when there is no in-point then the beginning of the clip is treated as the in-point). Two or more in-points can be set one after the other to shift them without creating a coding interval. This is not possible with out-points.

#### How is the interval between the in-point and the out-point coded?

At the same time the out-point is set, Videograph checks which category the user has currently activated in the coding bar. This supplies the coding value for the entire interval. The value overwrites all other values that are possibly already in the same interval (e.g. from earlier sessions). **Important:** even a zero status (not an activated coding category) will be evaluated and leads to the creation of an "empty" coding interval.

The consequence is that it is not necessary to have a continuous or uniform coding for the interval to be coded as only the status in the **last** second is what counts. It is, therefore, possible to make corrections until just before the out-point without having to go back to the in-point every time.

One variant results when the clip is paused before setting an out-point. The interval is then coded with the value that is shown just at that moment (i.e. second) in the **databar** of the timeline window. This value does not have to be from a current coding, it can be from an earlier one. This alternative can be applied when revising data at a later time, e.g. when an adjustment of intervals is to be made on the basis of available coding values.

An out-point **only** sets the coding interval for the currently **active** coding variable resp. its **active** databar. If you want to **use** this coding interval to code **further** variables, there are two alternatives: Either synchronous coding has been switched on, or better - because more comfortable- one works with special variables named "**master-variables**". This method is [described](#) in the next chapter.

#### Synchronous coding

When synchronous coding a group of variables is chosen which **jointly** affect the setting of in and out-points. It is then possible to code several variables parallel in uniform coding intervals. Setting out-points has no effect on variables that do not belong to the group. The command responsible for selecting variables is named "variables used in turns" and stands in the option menu of the coding window. It makes sense to use one of the variables as a criterium variable for when an interval change is to be made. The other variables should either contain only a single category or at least be so categorized that during the course of a coding interval never more than 1 category can apply.

## Typical proceeding

- a) Deactivate the playback interval and switch the write permission on.
- b) Set an in-point at the 0:00:00 position.
- c) Start and code the clip.
- d) Set an out-point at the place where the coding has to be changed. This creates simultaneously a new in point. When necessary, stop the clip before doing so.
- e) Let the clip continue to run and code new.
- f) Set new out point, etc.

## Navigation

The +/--buttons and the corresponding PAGE UP/PAGE DOWN keys function just like with every other coding without active playback interval, yet with one **extra**. When the **SHIFT**key is pressed, the clip always jumps to the current in-point, namely from every position before and after it.

## Further rules

- In order to be able to recognize where the current in-point is it is marked with a little red sentinel in the time ruler of the timeline window. All of the already existing coding intervals are additionally made visible by red borders in the data bars at the beginning and the end.
- The turn mode can be switched off by selecting the "turn mode off" command in the option menu. This removes (deletes) in-point and out-point and the red borders disappear when they have not been switched on by their own menu command.
- When you code while the clip is paused make sure that this is done at least 1 second after the in-point as always the elapsedsecond , i.e. the second to the left of the position pointer, will be coded.
- In synchronous coding depending on the number of variables an out-point might have to be set before one can code in all variables. From there it may be better to pause the clip before setting an out-point so the coding can be re-checked and - if necessary - be completed or changed.
- Even when coding is worked over, i.e. inserting new coding intervals or modifying existing ones at a later date you should pause the clip - especially with a synchronous procedure - before you set an out-point. The reason: when there is no recently activated coding category in a variable, then Videograph adopts the already existing value when setting an out-point instead of deleting it as would be the case when the clip is running. The best way to eliminate mistakes whenrevising is to switch the synchronous mode off and to work on the variables individually one after the other.
- When the "show data" option is switched on, allof the data shown in the data bars are reflected in the coding bars and shown as a colored category field so that optically nothing can be differentiated any more, i.e. whether it is a current or an old coding is. Whoever is bothered by this should switch off the option.
- When you activate the turn button without pressing CTRLat a position which is before the in-point, you will be asked whether you want to set a new in-point. If you say no, then your action is without effect.
- You can also begin to code before having set an in-point. When you push the turn button for the first time you will be asked whether you want to set an in-point or an out-point. If you select the out-point then the beginning of the clip is automatically the in-point.

## ► Using the coding window: The coding procedure

### Master-variables

A **master-variable** provides the **coding intervals** in the course of coding other variables, i.e. the **same** coding intervals are **created** as those that were used during coding the master-variable. In this way e.g. as many variables as desired can be coded using identical turns, without having to reenter the turns for each variable. It suffices to make a coding pass with one (guiding-) variable in order to produce a template for all further codings. This procedure has two advantages: It is not necessary to take care that the in-points and out-points are correctly positioned, and you can test the turns of the master-variable and correct them if necessary, before using them as template for further coding.

In principle any existing coding variable can be prepared for use as a master-variable. This is done simply by coding it, whereby the coding intervals are chosen in such a manner that they are valid for other variables as well. As a master-variable e.g. can serve a coding variable whose coding categories are able to provide a general classification schema, or a variable containing 1 or 2 dummy-categories that are only defined to mark coding intervals (turns).

How you have to code using a master variable:

- Activate the coding bar of variable X that you want to use as a master-variable.
- Open the "Options"-menu of coding window and choose the command "Use master-variable". You get a message indicating that now variable X is a master-variable and its coding intervals can be applied when coding other variables. At the same time the command "Use master-variable" in the "Options"-menu of coding window is replaced by the string "Master-variable is 'X'".  
Additionally - for better clarity - the databar of the master-variable in the timeline-window is placed on the top position and is locked here until the option "Use master-variable" is switched off or it is moved to another position by the user with the mouse.
- Activate the playback area, then the following functions: Always when (and only when) the + or - button of the coding window is pressed, you can go automatically backward or forward from one coding interval of the master-variable to the next. Then the first step is to bring the playback area by pressing the +/- buttons of the coding window to the interval where coding has to begin. Upon completing coding of this interval, go to the next interval to be coded by pressing the + (or -) button again. Technically spoken you are executing a normal time-sampling with varying, but fixed coding intervals.
- To switch off the master-variable, click the menu command again.

A master variable can be used as well as to define time intervals **during transcribing**.

## ► The coding procedure: practical tips

### Coding without an active playback interval

When you want to stop coding and begin again because the wrong category was chosen:

- Push the button -or the PAGE DOWN key. Then you jump to the begin of the coding. This can be done while the clip is running or when it is paused. If it is running it begins to run just after the jump. If it is paused, then it has to be re-started. Test what is easier for you. Remember while the clip is standing always the last second is coded, i.e. the second left of the position pointer.

You switched off a coding in a variable too late and want to correct this:

- Position the pointer at the place where the coding is to be deleted or overwritten. To overwrite it suffices to start the clip and newly code it. To delete you have to exactly place the playback interval on the "delete passage" and **activate** it. Then click twice on the category that is to be deleted. Do not forget to **deactivate** the playback interval before you continue.

I want to check all the codings again before exporting the data and correct them if necessary. What is the fastest way to do this ?

- Use the -/+ buttons or the PAGE DOWN/PAGE UP keys so you can go backward or forward from one coding to the next. The left border marker of the playback interval automatically moves to the coding begin and the right one to the end. To recode activate the playback interval and then click on the category field with the correct coding value. Before continuing you have to deactivate the playback interval, otherwise the control will not function any more.

Some codings end too early or too late? What is the fastest way to correct this ?

- Work with the method just described, but move the right border marker of the playback interval to where the coding is to end. It is possible that the marker cannot be moved. If this is the case it has to first be unlocked.

## ► Using the coding window

Clip control using  

As already described in the chapter about time sampling, using the two buttons shown here (placed in the toolbar of the coding window) the playback interval (including clip position) can be moved to the next or previous position without having to switch to the timeline window. There are also - depending on the context - some other types of clip control to which the two buttons can be used. Here is a description.

### Functions when the write permission is switched on

When the playback interval is **activated**, there are two variants:

#### "Normal" coding

This is the alternative described in the beginning. The playback interval is moved forward or backwards by 1 length; its size does not change.

#### Transcript-oriented coding

In this mode the playback interval is always moved to the beginning of the next transcription interval; whereby the following interval by definition always begins where the current one ends. Therefore, it makes no difference whether the next interval is transcribed or whether it is an untranscribed interval placed between two transcribed intervals. The playback interval also obtains the same length as the accessed transcription interval.

In areas where there are no intervals left to transcribe, then the functioning is the same as for "normal coding&".

**Illustration 1** makes this process more plastic.

#### Coding with a master-variable

In this mode the following playback interval automatically has the same size and position as the coding interval of the master-variable that lies next to the current playback interval (see **illustration 1**).

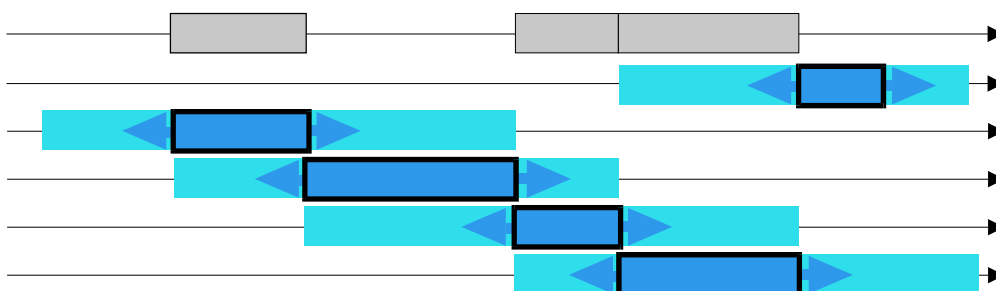
When the playback interval is **deactivated**, all the codings present in the active (i.e. red bordered) data bar can be accessed one after the other. The button **-** is used to go to the beginning of the preceding coding interval (related to the current clip position); with the button **+** the begin of following coding interval is reached. When there are no codings at all, the buttons have no effect. **Illustration 2** shows the principle of the procedure. Instead of the **+** button the PAGE UP key and instead of the **-** button the PAGE DOWN key can be used. Every action sets the left playback interval marker at the beginning and the right one at the end of the accessed coding interval. It is then possible - after a double click in the playback interval to activate it - easily and quickly to carry out recordings without having to replay the entire coding interval from beginning to end.

One modification results when the CTRL key is pressed and the current clip position is within a coding interval. The following or the previous coding is then not accessed, but rather the beginning or the end of the current interval.

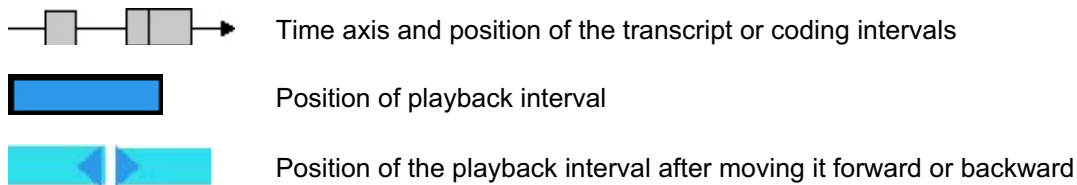
The control commands when the playback interval is deactivated do not just function in the coding window, they also function in the **timeline** window.

#### **Illustration 1: Control during coding:**

- on the basis of transcription intervals
- using a master-variable
- with activated playback interval without write permission



### Explanation:

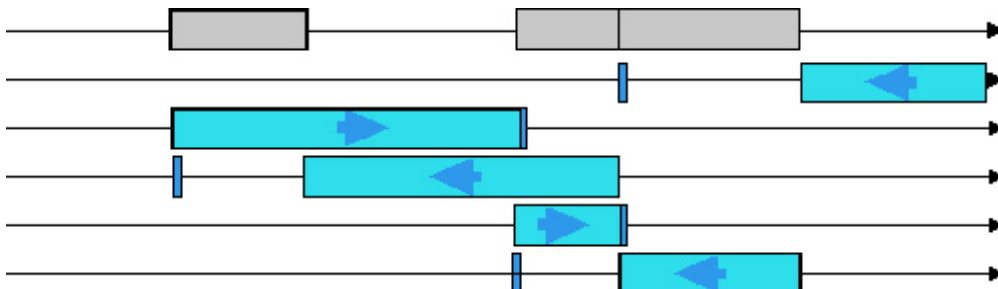


### Functions when the write permission is switched off

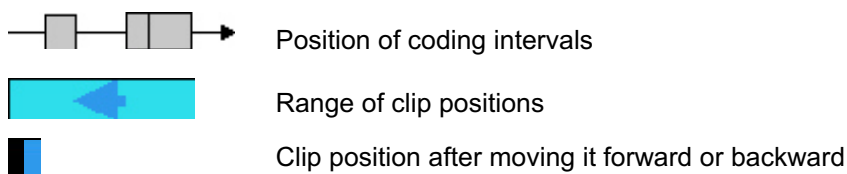
- When the playback interval is **activated** it depends on whether a databar is selected (recognizable by the red border) or not: when yes, the buttons function as shown in Illustration 1; when not, then - as in normal coding - the playback interval is moved forward or backwards by 1 length, whereby its size remains constant.
- When the playback interval is **deactivated**, the same thing happens as when the write permission is switched on (see above). A coding variable resp. a databar can be selected (by clicking on the head of the bar) and then moving the clip from one coding to the next as shown in Illustration 2. It is also possible to use the PAGE UP key instead of the button -; and instead of the button + the PAGE UP key can be used. With every action the left marker of the playback interval moves to the beginning and the right one to the end of the accessed coding interval. In this way it is possible - if the write protection is switched off and there is a double click in the playback interval to activate it - to simply and quickly recode without having to replay the entire coding interval from beginning to end.

One modification results when the CTRL key is pressed and the current clip position is within a coding interval. Then the beginning or the end of the current interval is accessed and not the following or the previous coding. The control commands with playback interval deactivated function not only in the coding window; they also function in the **timeline** window.

### Illustration 2: Control when the playback interval is deactivated



### Explanation:



The effect of the CTRL key is not shown in the picture.

## ▶ shortcuts in the coding window

Key	outcome
SPACEBAR	replay/pause
CURSOR UP	scroll coding bars (alternative: categories) upward
CURSOR DOWN	scroll coding bars (alternative: categories) downward
CURSOR RIGHT	wind forward clip by 1 second
CURSOR LEFT	rewind clip by 1 second
SHIFT + CURSOR RIGHT	wind forward clip to next tick mark
SHIFT + CURSOR LEFT	rewind clip to next tick mark
only if coding bar is <b>activated</b> :	
TABULATOR	move up following coding bar
TABULATOR+SHIFT	move down active coding bar
o otherwise:	
TABULATOR	activates timeline window
CTRL + M	set jump label at current time position
only if playback interval is <b>activated</b> :	
HOME (POS 1)	play from start of interval
CTRL + CURSOR RIGHT	move playback interval forward
CTRL + CURSOR LEFT	move playback interval backwards
only if databar is <b>activated</b> and playback interval is <b>deactivated</b> (see <a href="#">here</a> ):	
PAGE UP	wind forward to following coding interval
PAGE DOWN	Rewind to preceding coding interval
<b>CTRL</b> +PAGE UP or CTRL + CURSOR RIGHT	wind to end of current coding interval
<b>CTRL</b> +PAGE DOWN oder CTRL + CURSOR LEFT	wind to beginning of current coding interval
only when a <b>master-variable</b> is used:	
BILD OBEN	move playback area to following coding interval
BILD UNTEN	move playback area to preceding coding interval
only if <b>jump label(s)</b> set:	
NUMPAD*	move clip to preceding jump label
NUMPAD*	move clip to following jump label
only if there are <b>jump label(s)</b> and if playback area is <b>activated</b> :	
NUMPAD*	as above, but only if jump label lies within playback area
NUMPAD*	as above, but only if jump label lies within playback area
only if <b>write permission</b> and there are an <b>active coding bar</b> :	
KEYS 0-9 oder NUMPAD*0-9	numerical input of coding value
only if <b>write permission</b> and playback area is <b>deactivated</b> :	
'T'	insert out-point
CTRL+'T'	insert new in-point
debugging-function:	
ALT +'r'	rectify codingbar layout

NUMPAD = numeric keypad



## ► Transcribing in practice - step by step

### Introduction

When transcribing, a time interval is chosen from a clip (video or audio). This is played and the spoken contents are written down. Instead of paper there is a text window, the so-called transcription window, in which what has just been heard can be entered. Time entries are unnecessary as the program automatically measures time. The transcript can (including the time data) be written in an external file (exported), so that it can also be used for other programs and processed there.

The transcription procedure is supported by special program functions. Positions that need to be reached quickly in the clip, e.g. because something was difficult to understand, can be provided with a movable **jump marker** so that the push of a key suffices to reach the desired position. Ten of the twelve **function keys** can be allocated with the desired character string (called **autotext**) which can be entered in the transcript by pressing a key. In this way elements which repeatedly occur can be defined, e.g. automatic mnemonics for a change in speaker, so that having to type them in can be spared. Using the same mechanism the current video time can be added to the transcript by pressing a key. Furthermore, there are a number of **keyboard shortcuts** that can be used when transcribing the clip without having to noticeably interrupt the flow of the writing.

## ► Transcribing in practice - step by step

### Preparation

Before you begin transcribing the following pre-requisites must be created as videograph only allows entering the transcript text under the following conditions:

- The transcript window is open and activated. This is the case when a blinking writing marker (caret) is visible and in the title bar the text "transcript mediaclip ..." is shown. To activate the window click inside the window with the mouse.
- The transcription is then tied to the interval (time sampling procedure). In other words: A [playback interval](#) in the timeline window of the clip has to be chosen; one that sets the beginning and the end of the passage to be transcribed.  
The position pointer of the timeline window must also be within the playback interval (this condition is automatically fulfilled as long as the clip is running and only has to be paid attention to when wanting to transcribe while the clip is paused).
- Writing protection is rescinded. This happens when the button with the stylus symbol is pressed and locks in, changing its color to red. Note: the red color (instead of green) should be a warning and signal the lack of writing protection.
- Should several clips be open, the clip to be transcribed must be activated, e.g. you have to make sure that the the the transcription window and the timeline window of the clip to be transcribed are connected. (to see how it is done read [here](#)).

## ▶ Transcribing in practice - step by step

### Starting the transcription

Entering characters in the transcript window begins the transcription procedure. The scope of the text entered is **always** the **current** playback interval. Transcription is possible while the clip plays and while it is stopped. The second alternative may be necessary when the text is to be reworked or the transcription function is to be used for other purposes, e.g. when video scenes are to be furnished with commentaries. In the pause mode take care that the slider (position pointer) of the timeline window is in the playback interval (see above), since the text input is blocked. As long as the playback interval has not been changed (enlarged, reduced, relocated), the scope of the text in the transcript window remains constant and it does not matter if the text is changed, deleted, overwritten or replaced by a new text.

## ► Transcribing in practice - step by step

### Transcribing using constant intervals

This is probably the normal case: Transcribing takes place while the clip is running, whereby one begins with a playback interval that starts with the time position 0:00:00. The passage to be transcribed is repeated until the text has been completely entered and is without mistakes. When what has been heard outruns the transcription, the movie is stopped and the rest is written from memory, then it is restarted. Now and then you may have to return to passages that have already been listened to. At the end the playback interval is moved to the following time position, whereby the length of the interval can be maintained or varied. The game begins again and is repeated until the end of the clip has been reached.

Here are the practical instructions for how this scenario is to be handled technically.

- Set the left interval marker at the time position 0:00:00 and the right interval marker at the time position, that marks the end and/or the desired length of the playback interval. Then double click somewhere between the two markers with the left mouse key to activate the playback interval. Visible evidence of the activation is the color blue which the area between the two markers assumes.
- Start the clip, e.g. by clicking in the video window. There are several other ways of doing this, depending on which window is currently activated and can receive the input from the keyboard. When it is a matter of the timeline window you can use the <POS > or the SPACEBAR. When - following the instructions above - the transcript window has already been activated to be able to immediately begin with the input, then the two keys mentioned for text input are needed. The same is true for a number of other keys that fulfil a control function within the timeline window or the coding window, but are needed here for input purposes and whose functions have to be assumed by other keys. You can inform yourself about how the keys are assigned in the transcription window in detail in the table of [keyboard shortcuts](#) which offers a complete list of all of the key combinations. Yet back to the topic. There are two possibilities to start a clip (or to pause a running clip) within the transcript window. Either the key combination SHIFT + SPACEBAR is used or, what is probably easier, the <"0"> key on the numerical part of the keyboard (for units without a numerical part of the keyboard, use the CTRL + 'P' combination, see the keyboard table).
- Once the clip is running, check whether the blinking writing marker is visible and then begin to transcribe.
  - When you want to pause the clip for a moment, use the keyboard shortcut **NUMPAD '0'** (or **CTRL+SPACEBAR**) just mentioned. Use the same to continue.
  - The key combination **CTRL+'M'** is a very useful one. With it a **transcript marker** can be created at the exact time position of the clip. A transcript marker is a special kind of jump marker that is often important when you have to go back to certain positions to listen to them again and again. A transcript marker can always be moved to where it is needed while transcribing. You can learn more about this in the following.
  - It is possible to operate with several jump goals by setting "normal" jump markers (colored red) in the time ruler of the timeline window. The disadvantage of this method is that it requires two additional mouse clicks: In order to set a jump marker the timeline window has to be activated (mouse click 1), then the input focus has to be given back to the transcript window (mouse click 2). The activation of the jump markers can, on the other hand, also be effected in the transcript window using keyboard shortcuts. The chapter "[The timeline window](#)" described how standard jump markers are set and/or triggered.
  - Of course, you can manually set the position pointer to the second where you want to listen, either with the mouse in the time ruler or - without losing the input focus - with the keys **NUMPAD '4'** (backwards) and **NUMPAD '6'** (forward). The selection interval as a whole can also be moved using the same keys but pressing the **CTRL** key at the same time.

The clip automatically stops at the end of the playback interval.

If desired, you can begin at the beginning and replay the interval. This can be done by either manually resetting the position pointer or by using one of the following possibilities:

- a) You switched on the **loopmode** before starting the clip. The replay is automatically repeated after a 1 second delay. The button which switches the loop mode on and off is the one on the far right of the toolbar in the timeline window.
- b) Press the key combination **CTRL+SPACEBAR**, **NUMPAD '0'**, **NUMPAD ENTER** or **CTRL + NUMPAD ENTER**.

Before you continue with the transcription, check the status of the transcript window and activate it if the blinking writing marker is no longer visible.

Upon completing transcription of this interval, it is the next one's turn. Move the playback interval forward enough that there is no break between the two, but not so far that they overlap. You can do this by hand by first moving the right interval marker to the right edge with the mouse and then the left one to the left edge of the new playback interval. It is much better to use the **designated button**. It is a button marked with a plus in the toolbar of the transcription window. Every time it is pressed the interval moves forward so that its beginning is exactly where the end was (the button with the minus has - as is to be expected - the opposite effect). Another way to move the interval is to use the keyboard shortcut for it (see above).

## ► Transcribing in practice - step by step

### Transcribing event-controlled

With Videograph you do not necessarily have to always transcribe using the same time steps. You can decide when one transcription interval should end and another begin by using other criteria. This means that intervals of various lengths can result. In this connection we speak of "turns". One example is when a transcription interval depends on certain things happening within the material being viewed.

#### The principle of the procedure

You work interval-guided as in the normal transcription mode, i.e. there is an (activated) playback interval which is set at the beginning of the time phase to be transcribed. The difference is that the end is variable, i.e. a new transcription interval can be begun at any time, even when the end of the playback interval has not been reached.

#### What is a turn ?

A turn begins where the playback interval begins and ends when the turn button is pushed or at the end of the replay interval - whichever comes first.

#### Creating a turn

- As a turn can never be longer than the playback interval, the playback interval has to be at least as long as the turn to be created. There are three ways to do this.
  - Set the playback interval so that it is at least as long as the longest turn to be coded presumably lasts.
  - Should the length of the turn not be predictable, then stretch the playback interval until it reaches the end of the clip.
  - Not really recommendable: Lengthen the playback interval ad hoc when you notice the turn will go beyond its current end.
- Set the beginning of the playback interval at the place where the transcription is to begin. Start the clip and transcribe. All of the options that are open when transcribing with a constant time interval are available. Where a turn change is to take place, press the **turn button** which is on the far right of the toolbar in the transcription window. You can also use the **CTRL+T** key combination.
- Videograph then moves the playback interval to the end of the turn created from where you can directly continue with the transcription. The length of the interval is unchanged as long as there is enough room before the end of the clip. Reaching the end of the next turn click on the turn button, etc.
- Turns can be added but not inserted with the turn button, i.e. the playback interval has to be **after** all of the turns that have been created. You can only lengthen, shorten or insert new turns by proceeding as described in the chapter on "[rules and procedures for practical use](#)" under "modifying transcript intervals". No "empty" turns can be created. The turn button only functions when a text is entered.
- The **+** and **-** buttons (additional information [here](#)) can be used to "browse" back and forth through the completed transcription intervals, whereby the playback interval is automatically so placed that it covers the transcription interval exactly. In this way it is possible to change and re-write the transcript text at any time.

## ► Transcribing in practice - step by step

### Transcribing with a master variable

A master-variable is a coding variable which supplies the coding intervals for the coding of other variables. More detailed information how master-variables can be created and used during coding is given in the chapter "[Master-variables](#)" under the heading "The coding procedure". However a master-variable can not only be used to set coding intervals, it also is able to control the building of **transcription** intervals: then the coding intervals belonging to the master-variable **automatically** can be used as transcription intervals as well. Making use of this option makes sense when both transcript and codings are to be based on identical turns, because, thus, simply the same turns can be reused as often as you desires.

How you have to transcribe using a master variable:

- Declare a master-variable (see [here](#));
- **Activate** the transcription window and choose under the menu "transcript" the command "Use master-variable". You get a message indicating that now variable X is a master-variable and its coding intervals can be used to generate transcription intervals. At the same time the command "Use master-variable" in the "transcript"-menu of transcript window is replaced by the string "Master-variable is 'X'".
- **Activate** the playback area, then the following functions: Always **when** (and **only** when) the **+** or **-** button of the **transcription** window is pressed, the following transcription interval automatically has the same size and position as the coding interval of the master-variable that lies next to the current transcription interval. Then the first step is to bring the playback area by pressing the **+/-** buttons of the coding window to the position of the interval where transcribing has to begin. Upon completing transcribing this interval, go to the next interval to be transcribed by pressing the **+** (or **-**) button again. Technically spoken you are executing a normal transcribing with fixed, but variable sized time intervals.
- To switch off the master-variable, click the menu command again.

Please pay attention when the write permission is switched **on** and you choose a master-variable to try out it. The danger is that existing transcription intervals that are not identical with the coding intervals of the master-variable can be overwritten.

## ► Speaker-controlled transcription

### Procedures

An interesting variation is to create coding variables to be able to classify the transcribed intervals according to certain characteristics at the same time. They can either be coded parallel to the transcription or in another pass after transcription. For example, a variable can be defined to identify speakers or interactions and then coded according to who is speaking or who is speaking to who.

When working with a constant interval, the easiest thing to do is to code parallel to the transcription by just clicking back and forth between the coding and the transcript windows

When you work with turn, try out what is most comfortable.

- It is possible to code while the turn is running, but you must not forget to return to the transcript window before pressing the turn button. As is the rule in time sampling, the entire playback interval is also coded although the interval is usually longer than the turn. This may be confusing at the beginning, but it does no harm as the extra coding is either overwritten or deleted in the next turn.
- The turn is first closed and then you go back to code. The advantage is that the playback interval and the length of the turn are exactly the same. The disadvantage is that it could possibly be shortened, making it necessary to lengthen it to transcribe the next turn. This is easy with practice.
- It is really easy when the coding is done later in its own pass. You just have to switch on the "[use transcription intervals](#)" option in the options menu of coding window. Then the + and - buttons of the coding window can be used to move the playback interval (it must be activated naturally) from one transcription interval to the next and code it.



## ► Transcription with speaker coding: example

### Example

Here you see the illustration of a coding and transcript window with variables for coding which person is speaking and whether he/she is in the role of a sender or a receiver. "T" means "teacher" and "S10" e.g. the student with the number 10. Each student has his/her own variable. The window shown here has 20 student variables whereby the ones that are not visible can be made visible with Videograph's special [scrolling technique](#) with one click of the mouse in the scroll bar. Variables for 40 students are, thus, no problem.

When exporting, a data file with 41 numerical coding variables is created ( **Careful**: the commands for the "export of **codings**" have to be used - even when it is primarily a matter of transcription). This sounds rather voluminous, but it has two advantages:

- The data can be directly analyzed numerically, i.e. calculate the frequencies or conduct conjoint analyses.
- Per time interval or per turn as many speakers or speaker roles can be accessed as desired so there is no problem when they speak rapidly one after another or at the same time.

When you are only interested to record the current speaker and do not need the role information, then you only need 1 category per variable that is either clicked on or not.

P.S. Of course, you do not need to enter the definition information for each student variable. Just use the "open as template" command in the shortcut menu.

The screenshot shows the 'Codeview mediaclip 1' window. It features a toolbar with navigation and zoom controls. Below the toolbar is a grid of coding variables. Each variable is represented by a colored box with a label (e.g., 'T', 'class', 's1', 's2', etc.) and a table with two rows: '2 listener' and '1 speaker'. The 'T' and 'class' variables are highlighted in orange and pink respectively. The student variables (s1-s16) are arranged in two columns and are highlighted in various colors (teal, blue, orange, pink, green, yellow, light green, purple, light purple). Below the grid is a text area containing the text 'Here is written text'. At the bottom right, there is a time display showing '0:00:40 - 0:00:50'.

## ► Transcribing in practice - step by step

### Further methods

In videograph there is no restriction with regard to the length and time span of the transcription interval or the order in which they are worked on. You are free to choose where to begin the transcription, which time position to continue from and which length the interval is to have. It is also possible to transcribe while the clip is running or while it is paused. While paused you must make sure that the position pointer is within the transcription interval. The risk with such method is that an interval is selected without making sure it does not overlap with already transcribed intervals. When the new and the old intervals overlap, the new one overwrites the old one which means the section of time in the overlapping area has a new transcript. When the new interval completely overlaps the old one, the old one is deleted, when overlapping is partial, it is shortened timewise. The necessary adaptation of the transcription text then has to be done by the user. Whether there is an overlapping is easy to find out. This is done by searching the interval with the position pointer for existing transcriptions. Should the transcript window remain empty, then it is certainly an untranscribed time unit. Further information about overlapping intervals may be found in the section "Rules and procedures for practical use". Inform yourself here about the validity rules concerning executing transcription and about how to rescind an action if necessary.

## ► Transcribing in practice - step by step

### Autotext

Autotext consists of character strings, e.g. "teacher:" or "student:", that are defined and stored because they are needed often and you do not want to type them again and again. A key is pressed where they are needed and the desired string appears automatically in the text.

It is up to the user to decide wherefore an autotext string is defined, whether to mark a change in speaker or for other reasons.

#### How are text modules defined?

Using the "**autotext->define**" command in the "**transcript**" menu open a dialog window to enter the desired text. In the left column you find the key names **F2** to **F12** (without F10) which refer to the function keys in the upper most row of the keyboard. Each of these keys can be allocated with a string consisting of up to 50 alphanumeric characters. When the string is to appear in the text while transcribing, simply press the appropriate key. Insert point is always where the blinking cursor.

To the left of the column with the keys is one with checkboxes. These tell the program whether a carriage return is to be added before the string is inserted. The column on the far right also has such boxes. Here is where the program is told whether a carriage return is to be added following the autotext. You have to click on the checkbox where you want a new line. In the middle column is where the desired autotext is to be typed in. Simply leave the space empty when a key is not needed. When you are finished close the window. You can re-open it at any time to make any necessary changes.

#### How are text modules used?

There is not much to be said here. You simply press the corresponding function key. Autotext can be deleted or replaced by other, i.e. written over, at any time. There are also 3 strings that the system has **pre-defined** and that have a standardized use as long as they are not written over. These are the speaker terms "teacher:" and "student:", ( **F5** and **F6**) as well as the current time position of the clip (**F12**).

A autotext specially defined internally by the system is **\$VIDTIME\$**. When you allocate a function key with these characters (**capital letters** are a must), pressing this key the exact time position of the clip will be entered in the transcript. A combination of these characters and others can also be used, e.g. the string (**\$VIDTIME\$**) means that the time position is entered in parentheses.

#### How is autotext stored ?

Autotext elements are a component part of the project file and are also automatically stored. The disadvantage of this procedure is they are not available when a new project file is created, rather they have to be entered again. For this reason Videograph offers the additional possibility of storing them in a separate file from which the text can be imported when needed. Should you want to use this option, call the command "**autotext->save**" in the "**transcript**" menu and tell the program under which file name the current autotext strings are to be saved. This product is a normal plaintext file that can also be printed (but should not be changed) to use it as a master copy when transcribing.

#### How can stored autotext can be retrieved ?

When opening a project file, the autotext elements stored in it are automatically available. In order to use the autotext from a separate file, select the command "**autotext->load**" from the "**transcript**" menu. Please note: all existing autotext strings are **erased** when loading from a autotext file !

## ► Transcribing in practice - step by step

### Transcript marker

Using the **CTRL + 'M'** key combination, a **Transcript marker** can be created just at the time position where you enter text. A transcript marker is a special kind of jump marker that is required because it is often necessary to return to a specific position to listen to again. A transcript marker can be set where it is needed, or you can go back to where the marker is no matter whether the clip is running or is paused and at which time position the clip is in the playback interval without having to break the flow of the transcription since a unique keystroke suffices. It can be moved back and forth (with the keyboard or with the mouse) at will, deleted or when needed reset. This can take place parallel to text input without having to stop the clip. A transcript marker is recognized by its green color which differs from the red colored standard jump markers. The **CTRL + 'M'** key combination fulfills a double function: the transcript marker is created the first time it is used and the following times it is used the marker is moved to the current time position each time the key combination is pressed. The transcript marker is deleted with **CTRL+ENTF**. The opposite happens when using the **ENTER** key from the **numerical** part of the keyboard: the clip immediately jumps to the position of the marker and plays from there; if the beginning of the interval is desired, additionally the **CTRL-** key must be pressed. Using the '+' and '-' keys (**numerical** part of the keyboard) the transcript marker is moved one second forward or backward.

## ▶ Transcribing in practice - step by step

### Retro-window

When you advance from one transcription interval to the next while transcribing, it can sometimes be helpful when - in order not to lose the thought - there is a **retrospect** of the contents of the previous transcription interval. The **retro-window** is for this purpose. When the retro-window is open, not only the passage that is currently being worked on or that belongs to the current time position is visible, the text of the previous transcription interval is also visible (in the retro-window).

In order to **open** the retro-window, activate the transcript window so that the entry "**transcript**" appears in the menu bar. Then select the command "**Retrospection**". In order to close the retro-window, follow the same procedure.

The retro-window has a pink background to distinguish it from the input window. The height can be adjusted by pulling the lower edge of the window with the mouse to the desired position.

## ▶ Transcribing in practice - step by step

### Note-window

Into the note window additional text can be entered, e.g. notices and annotations relating to current transcript text. The same rules apply as for transcribing with the exception that text in the transcription window must be available before one can enter text into the note window. It is generally true that text in the note window is bound to the associated transcript text and the underlying time interval, e.g. when the transcript text is deleted then also the note. When exporting the transcript the notes are placed in an own data column following the transcript text.

In order to **open** the note-window, activate the transcript window so that the entry "**transcript**" appears in the menu bar. Then select the command "**note window**". In order to close the note-window, follow the same procedure.

The note-window has a yellowish background to distinguish it from the other windows. The height can be adjusted by pulling the lower edge of the window with the mouse to the desired position.

## ► Transcribing in practice - step by step

### Securing transcription intervals

When the writing protection is switched off and the playback interval is activated, it is not advisable to move the interval markers without a good reason, as this can change the length or the time span of the playback interval. This could lead to already transcribed time intervals being written over or modified (see below) because of the flexibility of the input procedure. There is a button in the toolbar of the timeline window that blocks the two interval markers in such a way that they cannot be reset individually. In order to relocate the interval, a designated button must be used. This a) always guarantees a constant interval length and b) guarantees that there is no overlapping between neighboring transcription intervals when relocating.

As long as you do not use variable, but constant time intervals, for transcribing, you should make use of the protective mechanism. Further details may be found in the description of the timeline window under "[securing playback intervals](#)".

## ► Transcribing in practice - step by step

### Storing the transcription

There are two time when storing data becomes important.

- **During transcription** to fix the individual steps:

With everything within the transcription process, whether it is the transcription of time intervals or changes in the text or any other manipulations, make sure that these actions first become **valid** when one of the following happens:

- The fix button is pressed (only functions when clip is stopped).
  - The clip is stopped **or** briefly started, depend on whether transcription took place while clip was running or was stopped
  - The position pointer goes outside the playback interval or the playback interval is displaced using the +/- buttons.
- When this condition is not fulfilled, the changes made are wasted.
  - **At the end of the program** when it is a matter of filing what has been done until then:  
The transcript is part of the project file and needs no special storage. It suffices to store the project file. The current status is always stored so that the transcription can cover several working sessions without interruption. When the project file is loaded, the transcription can be continued from where the previous session ended. An external possibility of documentation is the **export** in an **SPSS database** or in a tab-delimited **text file** in purely ASCII format. This procedure is described [elsewhere](#).



## ► Transcribing in practice - step by step

### The importance of writing protection

The write protection signals the program the current processing mode, i.e. it shows in which context a certain user action is taking place: when it is switched off, the transcription phase is running; when it is on, the user is doing something else. The write protection, thus, fulfils an important task: it prevents changes in the transcription or its intervals that could be caused by actions or circumstances outside the transcription mode. Especially the playback is a origin of danger as it fulfils diverse functions both within and without the transcription mode. In the transcription mode it serves to build transcription intervals, in the coding mode it sets the time units for coding, and elsewhere it is used for selective replay of individual passages. Think of the adulterated or even destructive effect it could have when the **write protection is switched off** and a playback interval of any size and length is chosen and activated in the area where one or several transcribed intervals are and the clip is then started: upon completion of the replay everything within the chosen area that has already been transcribed is either written over or deleted. To avoid such calamities, make it a habit to work with the **write protection switched on** unless transcribing.

## ► Transcribing in practice - step by step

### Creating a new transcription interval

This procedure only functions when the write protection is switched **off**.

Mark the area of time that is to be transcribed by setting and activating the playback interval. Otherwise follow the instructions in the chapter "Preparations" above. Start the video clip and begin the text input in the transcript window. Should the playback not be needed, you can also transcribe without the clip running. You must, however, in this case make sure that the position pointer is in the playback interval.

The conclusion of the procedure is a signal which attests the validity of the transcription. No new transcription interval can be realized before this signal has not be given. A signal shows when one of the following conditions occurs or is created:

- The fix button is pressed (only functions when the clip is not running).
- The clip changes modes, i.e. it changes from replay to the stopped position **or** vice versa, depending on whether transcribing took place while the clip was running or stopped.
- The position pointer goes outside the playback interval.
- The playback interval is moved using one of the two buttons for that purpose.

Visible proof of the newly created transcription interval is that the beginning and the end of the interval are shown in the status bar of the transcript window. When you realize you have done something wrong and want to discontinue the transcription without creating a transcription interval, there are two possibilities: **before** one of the conditions above becomes reality either deactivate the playback interval or switch off the write permission.

A third possibility would simply be to later delete the interval created.

## ▶ Transcribing in practice - step by step

### Modifying transcribed text in a transcription interval

This procedure only functions when the write protection is switched **off**.

Place the position pointer within the interval so that the text to be edited appears in the transcript window and make the changes. To do so it is **not** necessary to set and activate a playback interval. It makes more sense when the clip is not running.

Before videograph adopts the changes you are asked to confirm with YES (= adopt changes) or NO (= ignore changes). Should the changes become affective direct and without asking, push the fix button (only works when the clip is paused). Should the changes be canceled without being asked, switch off the writing permission.

## ▶ Transcribing in practice - step by step

### Deleting an existing transcription interval

This procedure only functions when the writing protector is switched **off**.

A congruent playback interval first has to be created, i.e. set the left interval marker at the beginning and the right one at the end of the transcription interval. Place the position pointer within the interval and activate it. Then delete the entire text in the transcript window. This can be done while the clip is running or stopped. The deletion is (irrevocably) effected as soon as one of the following happens:

- The clip is - depending on its state while deleting - either stopped or started for a moment.
- The position pointer leaves the interval to be deleted.
- The fix button is pressed (only functions when the clip is stopped).

The deletion is invalidated when the playback interval is deactivated or permission to write is switched off before one of these conditions has been fulfilled.

Several intervals can be deleted this way at the same time.

## ► Transcribing in practice - step by step

### Expanding or shortening an existing transcription interval

This is also possible. The location and time span of an existing transcription interval can be changed retroactively, thus expanding or shortening the area of validity for the text contained in it. The procedure only functions when the writing protection has been switched **off**. First create a matching playback interval, i.e. set the left interval marker at the beginning and the right one at the end of the transcription interval to be changed. Place the position pointer in the interval and activate it. Then pull the left marker to the left and/or the right one to the right until the interval has the desired length or time span. Between the new and the old interval there must be a joint area and the position pointer must be in this overlapping area. To guarantee this the clip should be paused. The change becomes affective as soon as one of the following conditions is fulfilled:

- The clip is started for a moment and then paused.
- The fix button is pressed (clip has to be in the stop mode).

The change is invalidated by either deactivating the playback interval or switching off the writing permit before these conditions become affective.

#### **Secondary effects:**

- The remaining interval that is left over when the interval is shortened is not deleted, it is separated. It is transformed into a new interval including the text.
- When other existing transcription intervals overlap because of the expansion or relocation, these are then shortened correspondingly. The extreme example would then be the complete loss of an interval.

## ▶ Transcribing in practice - step by step

### Choosing a font

The default font for the transcript text is the 8 pt. "**MS Sans Serif**". Whoever prefers can select another typeface or other style characteristics by means of the command "**font...**" in the transcript window. One possible use is for presentations at conferences where a larger font size is needed than for transcribing on the monitor. The choosed font is kept im mind by videograph (by creating a registry key) and restored when the next session starts.

## ▶ Transcribing in practice - step by step

### Using the clipboard

When you have copied text into the Windows clipboard, you can paste it anywhere and as often as you want in the transcript. It makes no difference whether the text is from an external application or from the transcript itself. As there is no menu command for it, you have to use a keyboard shortcut:

- To insert text from the clipboard, press **SHIFT+Insert**. The insert position is always the current position of the cursor. This only functions when permission for writing is given and the playback interval is activated.
- In order to copy text from the transcript into the clipboard, mark it and then press **CTRL+Insert**.
- In order to delete (cut) text and copy it into the clipboard, mark it and then press **SHIFT+Delete**.

## ▶ Transcribing in practice - step by step

### Clip control with

To avoid having to switch back and forth between the timeline window and the transcript window while transcribing when the clip is to be moved to another position, there are the two buttons shown above. Several kinds of control are possible with them

#### Functions when the write permission is switched on

Here the playback interval must be **activated**. Otherwise the buttons have no function. Then there are two variants:

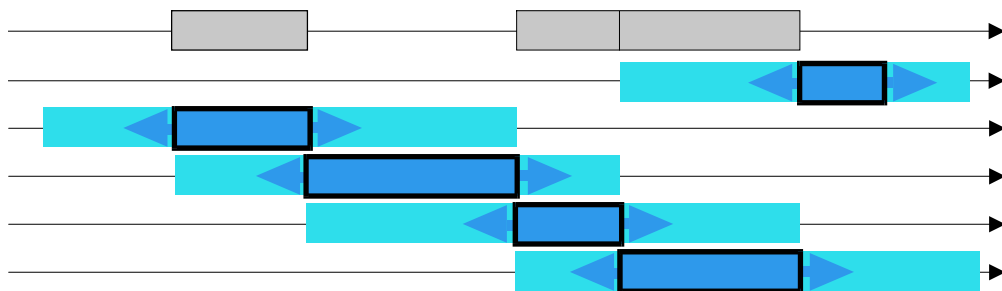
##### a) "Normal" transcribing

When **no** transcribed intervals follow the playback interval, then the playback interval (and thus the clip) is pushed forward by 1 length and its size does not change. This is the typical case with successive transcription of time intervals of the same size, i.e. the usual procedure. The same principle holds true for backward movement. Should there be time segments to be transcribed before or after the playback interval; whereby this always begins by definition where the current one ends. It, therefore, makes no difference whether the interval is transcribed or whether it is an untranscribed interval that is between two transcribed intervals. The playback interval also obtains the **same length** as the accessed time interval. Illustration 1 demonstrates the principle using examples. This type of access can be used to easily check the created transcription intervals later or in between and make changes if necessary.

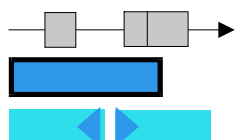
##### b) Transcribing with a master variable

In this mode the following transcription interval automatically has the same size and position as the coding interval of the master-variable that lies next to the current transcription interval.

#### Illustration 1: Clip control with activated playback interval and write permission is switched on (with oder without a master-variable)



#### explanation:



Time axis and position of the transcription intervals

Position of playback interval

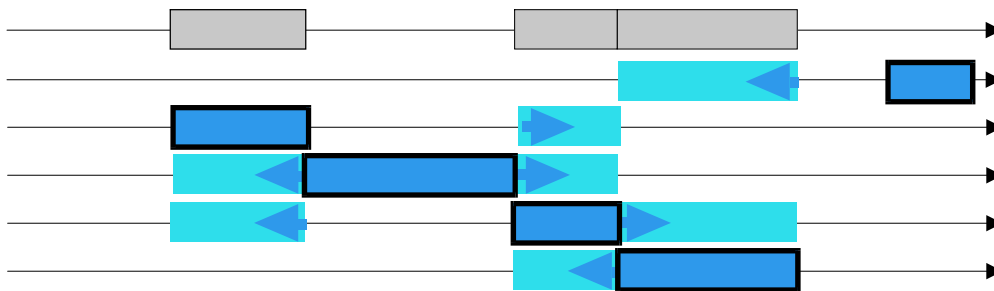
Position of playback interval after moving it forward or backward



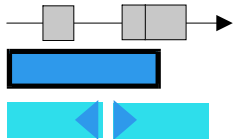
## Functions when the write permission is switched off

- When the playback interval is **activated**, the buttons function as they do when the permission to write is turned on (Illustration 1).
- When the playback interval is **deactivated** it is possible - as shown in Illustration 2 - jump directly from one transcribed interval to the next, so that it is possible to scroll through the transcript without manual navigation being necessary. When you want to amend the text, turn the permission to write on, make the change and push the fix button. Then turn the permission to write off again so you can scroll. When the position pointer stands outside of the playback area, the target position after jump is determined by the **position of pointer** and not by the position of playback area. When the position pointer stands inside a transcription interval and you don't want to jump but just to fetch the playback area, you must press SHIFT while clicking a button.

**Illustration 2: Clip control when the playback interval is *deactivated* and write permission switched *off***



### explanation:



Time axis and position of the transcription intervals

Position of playback interval

Position of playback interval after moving it forward or backward

► **Keyboard shortcuts in transcription window**

key(s)	alternatively	action
F5 (reserved but modifiable)		Inserts a carriage return at the caret position followed by the string "teacher:"
F6 (reserved but modifiable)		Inserts a carriage return at the caret position followed by the string "student:"
F12 (reserved but modifiable)		Inserts (at the caret position) a string containing the current video time
F2 - F12, beside F10		Insert a user-defined string at the caret position
<b>CTRL+ 'M'</b>		use to set or drag a <b>transcription marker</b>
CTRL+NUMPAD '+'	SHIFT+CTRL + '+'	move clip to proximate jump label
CTRL+NUMPAD '-'	CTRL +SHIFT + '-'	move clip to preceding jump label
only if <b>transcription marker</b> set		
CTRL+ENTF		erase <b>transcription marker</b>
NUMPAD '+'	CTRL + '+'	move forward (inside the playback area) <b>transcription marker</b> by 1 second
NUMPAD '-'	CTRL + '-'	move backward (inside the playback area) <b>transcription marker</b> by 1 second
NUMPAD ENTER	CTRL + 'H'	play from <b>transcription marker</b> (if existing)
CTRL+NUMPAD ENTER	SHIFT+ CTRL+ 'H'	play from start of interval
only if playback area is <b>activated</b> :		
CTRL+SPACE		play/pause
NUMPAD '0'	CTRL + 'P'	play/pause
NUMPAD ENTER	SHIFT+ CTRL+ 'H'	play from start of interval
NUMPAD '6'	CTRL + 'F'	wind forward clip by 1 second
CTRL+NUMPAD '6'	SHIFT+ CTRL + 'F'	move forward playback area
NUMPAD '4'	CTRL + 'B'	rewind clip by 1 second
CTRL+NUMPAD '4'	SHIFT+ CTRL + 'B'	shift back playback area
only <b>inside</b> of active playback area:		
NUMPAD ENTER	CTRL + 'H'	play from start of interval ( <b>or</b> play from transcription marker if existing)
NUMPAD '0'	CTRL + 'P'	play/pause
CTRL+NUMPAD '+'	SHIFT+ CTRL + '+'	go to proximate jump label inside the playback area
CTRL+NUMPAD '-'	CTRL +SHIFT + '-'	go to preceding jump label inside the playback area
only if <b>write permission</b> :		
CTRL+'T'		begin new turn
only if write <b>permission</b> and playback area <b>activated</b> :		
SHIFT+Einfg		Insert text from clipboard at the caret position
CTRL+Einfg		Copy selected text to the clipboard
SHIFT+Entf		Cut selected text and copy it to the clipboard

NUMPAD = numeric keypad

## ► Data import and data export

### Importing data bars

In the chapter "[Importing coding variable](#)" it was described how variable definitions can be added to a videograph file, i.e. by importing them from another videograph file. The same is also possible with **data bars**. By means of a simple import procedure, data bars from another project file (\*.VDG) can be added to a project file. In this way, e.g., it is possible to bring data from various coders and spread across different files together for comparison purposes in one file.

The menu commands for importing data bars are found both in the menu of the timeline window under "Import/Export" and in the menu of the coding window under "coding".

Please note that a data bar can **only** be imported when there is a **coding variable** with the **same** name and the **same** value range as the coding variable to which the data bar in the original file belongs.

There are two methods to import. Method 1 is when a coding variable is **selected** (recognizable by the red frame, see [here](#) for the choice of coding variables). In this case only the data bars whose coding variable is identical with the chosen one can be imported. In the other case, when **no** coding variable has been chosen (Method 2), all of the data bars whose affiliation to the coding variable is identical to one of the coding variables already present in the current project file can be imported.

The first step when importing is to select the method. If method 1 is the one, a coding bar is selected. If method 2 is chosen, make sure there is no selection available. Then choose the command from the menu and indicate from which file is to be imported. After videograph has verified the chosen import file, the names of the data bars suitable for import are shown in a dialogue window; or if the file is completely unsuitable for import, an error message appears.

The dialogue window is subdivided into three areas: in the upper left window (area 1) are the data bars that can be imported direct; the lower left window (area 2) shows data bars that have a name that has already been used and therefore have to be renamed before they can be imported. The choice of which data bars in area 1 (one, several or all) actually ought to be imported is made by relocating the desired elements from area 1 in area 3 (right window). There is also an information button which can give a description of all of the data bars shown as well as the names of the coding variables and which data bar they belong to.

## ▶ Help on import dialogue

The **upper left window** contains the list of **importable** data bars, in the **right window** the names of the data bars that are to be **imported**. The lower left window shows data bars that cannot be imported because they have a name that has already been used. In this case you can rename the data bar.

**Using** the mouse, select the data bars from the **left** list that you want to import.



Click here to move the selected data bars from the **left** to the **right** window.

You can also **reverse** the procedure. Select a data bar that you want to **delete** from the import list from the **right** window.



Click here to move the selected data bar from the **right** window to the **left** one.



A data bar that cannot be imported can be made importable by renaming it. Select a data bar and click here to rename it.



Select a data bar and click here for its description. Once the information window has been opened you can continually recall information about further data bars without having to close the window beforehand.

## ► Exporting coding data

### Basic rules

The commands in the menu for exporting data to SPSS can be found both in the menu in the timeline window under "Import/Export" and in the menu of the coding window under "Coding". The SPSS file name can be defined in a dialogue window and the options for export chosen. The functions of the options will be treated in the next two sections. When choosing a name for the SPSS file, make sure that the file name ends with **.SAV**.

In the method described here it is not possible to add data to existing files (due to the restrictions within SPSS). When there is already a file of the same name, then it is written over. When you do not want to create a new SPSS file, but want to insert the coding as a variable in an existing file, you have to use another method. The other method is described in the chapter "Exporting data as textfile".

When only 1 video clip is loaded, it makes no difference which menu is used for exporting. Otherwise: When using the export command from the timeline window menu, only coding data that belongs to the activated timeline window can be exported. Using the same command from the menu in the coding window, the coding data of all of the available timeline windows can be exported.

When exporting coding data, the user selects the one he/she wants to export from the **data bars** available in the timeline window. These are then transformed into SPSS variables and written in an SPSS file \*.SAV.

The first variable in the file is the **name** of the clip including the path. The variable is a STRING type and can take up to 255 characters. The variables with the coding data then follows. A choice between two options can be made for every data bar:

- The data bar is represented in **three** SPSS variables that form a group and are always in the same order.
  - a) The specification of the time for the **beginning** of the coding interval. The variable is a time variable in hh:mm:ss format.
  - b) The specification of the time for the **end** of the coding interval. The variable is a time variable in hh:mm:ss format.
  - c) **Coding values** for the intervals. The variable is numerical (F3.0).
- The time variables show the time position in the video (starting time = 00:00:00).
- Only **one** SPSS variable is created with the coding **values**, the coding intervals are not transferred. This procedure is practical when the data bars have identical coding intervals.

When deciding on names for the variables, the following principle is used:

1. The name of Variable 1 (begin of the interval) and Variable 2 (end of the interval) consists of a prefix followed by a serial number for the videograph coding variable to which the exported data bar belongs and a serial number for the data bar itself. The prefix is **t1\_** for Variable 1 (beginning of the interval) and **t2\_** for Variable 2 (end of the interval). Examples of names are: **t\_01\_02**, which means that it is the second data bar of a coding variable, **t\_02\_01** etc.
2. The name of Variable 3 (coding value) is identical with the name of the data bar in videograph.

The Variable 1 and 2 names are purely formal letter combinations to characterize their belonging together, they have no other meaning.

The category names of the data bars and the label given the bar also are exported. They are assigned to variable 3 as value labels resp. variable labels. Empty data bars without any data cannot be exported.

The data in each data bar are written as cases in the same sequence in the SPSS file in which they come timewise in the data bar, i.e. the variables 1 (beginning of the interval) and 2 (end of the interval) form an ascending time serie. Otherwise the file structure depends on how many codings exist for each variable, on whether constant time intervals were used in coding, whether there are any uncoded intervals, and which of the options offered by videograph for exporting data were used. The following tables show prototype data structures:

## ► Exporting coding data

### Export with variable coding interval with and without residual interval

The following extract from an SPSS file contains the exported data from 2 data bars V1 and V2 (for lack of space columns 1 and 5 do not show the filename of the clip; instead there is a place holder). The data structure is based on the following coding structure (the variables V1 and V2 are characterized as variables in the following for the purpose of simplicity):

**Differing coding intervals** were used - both within the data bars and between them. Furthermore, there were **uncoded** time intervals in both variables. This is due to the fact that **no residual category** like e.g. "the teacher does something else" has been defined for the case that the other categories do not apply.

**Table 1**

FIL02_01	t1_02_01	t2_02_01	V1	FIL01_01	t1_01_01	t2_01_01	V2
Video	0:00:00	0:00:13	8	Video	0:00:00	0:00:15	2
Video	0:00:31	0:00:38	7	Video	0:00:27	0:00:44	1
Video	0:00:49	0:01:07	4	Video	0:00:44	0:00:55	2
Video	0:01:15	0:01:24	4	Video	0:01:08	0:01:25	1
Video	0:01:28	0:01:35	7	Video	0:01:45	0:01:55	2
Video	0:01:35	0:01:53	1	Video	0:02:01	0:02:20	1
Video	0:02:05	0:02:10	2	Video	0:02:39	0:02:45	2
Video	0:02:20	0:02:30	6	Video	0:03:03	0:03:09	2
Video	0:02:41	0:02:59	8		,	,	,
Video	0:03:13	0:03:15	8		,	,	,

As you can see, the time interval forms an ascending time serie, but the serie is **discontinuous** because there are uncoded intervals. As the uncoded (and coded) intervals in variable V1 are not congruent with those of V2, the time series for V1 and V2 are **not congruent**. Whether this circumstance is irrelevant or disadvantageous depends on the analysis method planned.

There are two possibilities to get a time serie without gaps:

- Avoid uncoded intervals by defining a residual category like "the teacher does something else". This possibly increases the time needed for coding considerably as there have to be as many residual categories as there are coding variables.
- An option offered by videograph for exporting data is used to create a time serie without gaps when exporting without having to define residual categories. The option is to be found in the dialogue box for data export in the form of a checkbox and is named "also store uncoded intervals". It causes Videograph to code intervals with missing variable value (coding gaps) before export with the SYSMISS value. The SPSS data then forms a continuous time serie without having to define a residual category for this reason. The SYSMISS value can later be recoded in any other value within SPSS depending on what the analysis is for.

The following Table 2 shows - using the same basic coding structure - the first 10 cases of the SPSS data structure for coding bars V1 and V2, when the "save uncoded intervals too" option is used for export. Note the lack of parallelism between the time series for V1 and V2 that still exists. This potential problem can be solved by using the same constant coding interval for all variables (see Tables 3 and 4).

**Table 2**

FIL02_01	t1_02_01	t2_02_01	V1	FIL01_01	t1_01_01	t2_01_01	V2
Video	0:00:00	0:00:13	8	Video	0:00:00	0:00:15	2
Video	0:00:13	0:00:31	,	Video	0:00:15	0:00:27	,
Video	0:00:31	0:00:38	7	Video	0:00:27	0:00:44	1
Video	0:00:38	0:00:49	,	Video	0:00:44	0:00:55	2
Video	0:00:49	0:01:07	4	Video	0:00:55	0:01:08	,
Video	0:01:07	0:01:15	,	Video	0:01:08	0:01:25	1
Video	0:01:15	0:01:24	4	Video	0:01:25	0:01:45	,
Video	0:01:24	0:01:28	,	Video	0:01:45	0:01:55	2
Video	0:01:28	0:01:35	7	Video	0:01:55	0:02:01	,
Video	0:01:35	0:01:53	1	Video	0:02:01	0:02:20	1

## ► Exporting coding data

### Exporting with constant coding intervals with and without residual coding

The following extract from an SPSS file contains the exported data from 2 data bars V1 und V2 (for lack of space columns 1 and 5 do not show the filename of the clip; instead there is a place holder). The data structure is based on the following coding structure (the variables V1 and V2 are characterized as variables in the following for the purpose of simplicity):

Both within a data bar and between them a **constant** coding interval of 10 seconds was used. There were also **uncoded** time intervals in both variables. This is due to the fact that **no residual category** like "the teacher did something else" was defined for the case that the other categories did not apply. The length of the uncoded time intervals **varies**, yet (with the exception of the last remaining interval not shown here) it always has a whole numbered **multiple** of 10 seconds because of the constant coding interval.

The export takes place with the "save uncoded intervals too" option. Due to the option, the SPSS data form a time serie **without gaps** even when there are uncoded time intervals. Residual categories are **unnecessary** to avoid uncoded intervals. Further information may be obtained from the previous section.

**Table 3**

FIL02_01	t1_02_01	t2_02_01	V1	FIL01_01	t1_01_01	t2_01_01	V2
Video	0:00:00	0:00:10	6	Video	0:00:00	0:00:10	3
Video	0:00:10	0:00:20	,	Video	0:00:10	0:00:30	,
Video	0:00:20	0:00:30	7	Video	0:00:30	0:00:40	5
Video	0:00:30	0:00:40	,	Video	0:00:40	0:01:00	,
Video	0:00:40	0:00:50	1	Video	0:01:00	0:01:10	2
Video	0:00:50	0:01:00	,	Video	0:01:10	0:01:30	,
Video	0:01:00	0:01:10	1	Video	0:01:30	0:01:40	3
Video	0:01:10	0:01:20	,	Video	0:01:40	0:02:00	,
Video	0:01:20	0:01:30	6	Video	0:02:00	0:02:10	3
Video	0:01:30	0:01:40	,	Video	0:02:10	0:02:30	,

As you can see, the time intervals in both variables form a **continuous** time serie. What is missing is parallelism between the variables. The reason is that the uncoded intervals are not all the same length. Whether this is desired or not depends on the planned method of analysis.

There are two possibilities to get identical time series:

- Avoid uncoded intervals by defining a residual category like "the teacher does something else". This possibly increases the time needed for coding considerably as there have to be as many residual categories as there are coding variables.
- An **option videograph offers for exporting data** is used to create identical time series when exporting **without** having to use residual categories. The option is to be found in the dialogue box for data export in the form of a checkbox and is named "use selected play area". It causes Videograph to divide, if necessary and possible, all of the uncoded intervals in sections of **equal length** like the **active playback area** before exporting the data and to code each section with the SYMISS-value. When, for example, the active playback interval is 15 seconds long and an uncoded interval is 30 seconds, then the uncoded one is divided into two parts of 15 second lengths; an uncoded interval of 42 seconds results in 3 intervals of 15, 15, and 12 seconds; and uncoded intervals of 15 seconds or less stay the way they are.



When the exported coding data are supposed to form parallel time series between the variables, then the following pre-requisites have to be fulfilled:

- The exported intervals have to form a continuous time serie. To guarantee this, the "save uncoded intervals too" button has to be marked in the dialogue box for data export.
- All coding is carried out with a constant coding interval (e.g. 10 seconds) and at the time of export the playback interval of the same length as the coding interval is activated (also 10 seconds).
- The "use selected play area" option is checked for data export in the dialogue box. This possibility only exists when an activated playback interval exists and at the same time the "save uncoded intervals too" box is checked.

When these conditions are fulfilled, all of the time series - with reference to the sample table - in the SPSS file are made of time intervals of 10 seconds which follow each other 10 seconds apart.

Table 4 shows - using the same basic coding structure as Table 3 - the first 10 cases of the SPSS data structure for data bars V1 and V2, when the data are exported making use of the options "save uncoded intervals too" and "use selected play area". Note the total congruence between the time series for V1 and V2.

**Tabelle 4**

FIL02_01	t1_02_01	t2_02_01	V1	FIL01_01	t1_01_01	t2_01_01	V2
Video	0:00:00	0:00:10	6	Video	0:00:00	0:00:10	3
Video	0:00:10	0:00:20	,	Video	0:00:10	0:00:20	,
Video	0:00:20	0:00:30	7	Video	0:00:20	0:00:30	,
Video	0:00:30	0:00:40	,	Video	0:00:30	0:00:40	5
Video	0:00:40	0:00:50	1	Video	0:00:40	0:00:50	,
Video	0:00:50	0:01:00	,	Video	0:00:50	0:01:00	,
Video	0:01:00	0:01:10	1	Video	0:01:00	0:01:10	2
Video	0:01:10	0:01:20	,	Video	0:01:10	0:01:20	,
Video	0:01:20	0:01:30	6	Video	0:01:20	0:01:30	,
Video	0:01:30	0:01:40	,	Video	0:01:30	0:01:40	3

## ► Exporting coding data

### Exporting of codings without or with variable time interval

Incongruent coding intervals are created between the coded variables inevitably with these two coding methods. When the results are then transferred to an SPSS file without being transformed, time series result that are not necessarily exactly the same between the variables. This means the data structure is not especially suited for multivariate analysis. Export **Type III** is there to avoid this. The entire coding period (i.e. both the coded and the uncoded passages) is so subdivided by Videograph before export that identical time intervals are created for all variables. Thus, there are only identical time series in the export file, but with intervals of various lengths within a time series, whereby the shortest interval can last only 1 second.

To select Export Type III, simply check the "parallelize intervals" box in the [export dialog](#).

The following extract from an SPSS file contains the exported data for coding variables V1 and V2 **without** the use of a prescribed coding interval ("event sampling").

**Tabelle 5**

t1_02_01	t2_02_01	V1	t1_01_01	t2_01_01	V2
0:00:00	0:00:01	,	0:00:00	0:00:18	4
0:00:01	0:00:16	7	0:00:18	0:00:25	,
0:00:16	0:00:17	,	0:00:25	0:00:43	6
0:00:17	0:00:38	5	0:00:43	0:00:53	2
0:00:38	0:00:45	,	0:00:53	0:01:01	,
0:00:45	0:01:05	6	0:01:01	0:01:42	1
0:01:05	0:01:17	,	0:01:42	0:02:29	5
0:01:17	0:01:38	3	0:02:29	0:03:12	3
0:01:38	0:01:41	,	0:03:12	0:03:32	,
0:01:41	0:01:52	6	0:03:32	0:04:10	2
0:01:52	0:02:10	,	0:04:10	0:04:50	6
0:02:10	0:02:25	2	0:04:50	0:04:54	5
0:02:25	0:02:36	,	0:04:54	0:04:59	4
0:02:36	0:02:53	2	0:04:59	0:05:03	3
0:02:53	0:03:04	,	0:05:03	0:05:09	2
0:03:04	0:03:09	4	0:05:09	0:05:17	1
0:03:09	0:03:24	,	0:05:17	0:05:31	2
0:03:24	0:03:29	7	0:05:31	0:05:55	3
0:03:29	0:03:45	,	0:05:55	0:06:30	4
0:03:45	0:04:00	1	0:06:30	0:07:11	6
0:04:00	0:04:08	,	0:07:11	0:07:55	2
0:04:08	0:04:15	4	0:07:55	0:08:04	1
0:04:15	0:04:27	,	0:08:04	0:08:05	,
0:04:27	0:04:41	8	0:08:05	0:08:28	2
0:04:41	0:04:47	,	0:08:28	0:09:33	5
0:04:47	0:04:58	5	0:09:33	0:10:13	3
0:04:58	0:05:04	6	0:10:13	0:10:32	,
0:05:04	0:05:13	,	0:10:32	0:10:48	3

The following extract from an SPSS file is based on the same coding data as above. The data, however, were exported using Export Type III. Note the complete congruence of the time series for V1 and V2. One disadvantage of the method that cannot be overseen - besides the varying lengths of the individual intervals within a time series - is the potential minimalizing of the intervals and the connected increase in cases.

**Tabelle 6**

t1_02_01	t2_02_01	V1	t1_01_01	t2_01_01	V2
0:00:00	0:00:01	,	0:00:00	0:00:01	4
0:00:01	0:00:16	7	0:00:01	0:00:16	4
0:00:16	0:00:18	,	0:00:16	0:00:18	4
0:00:18	0:00:19	,	0:00:18	0:00:19	,
0:00:19	0:00:25	5	0:00:19	0:00:25	,
0:00:25	0:00:35	5	0:00:25	0:00:35	6
0:00:35	0:00:38	5	0:00:35	0:00:38	6
0:00:38	0:00:43	,	0:00:38	0:00:43	6
0:00:43	0:00:45	,	0:00:43	0:00:45	2
0:00:45	0:00:53	6	0:00:45	0:00:53	2
0:00:53	0:01:01	6	0:00:53	0:01:01	,
0:01:01	0:01:03	6	0:01:01	:01:03	1
0:01:03	0:01:05	6	0:01:03	0:01:05	1
0:01:05	0:01:14	,	0:01:05	0:01:14	1
0:01:14	0:01:19	,	0:01:14	0:01:19	1
0:01:19	0:01:20	3	0:01:19	0:01:20	1
0:01:20	0:01:33	3	0:01:20	0:01:33	1
0:01:33	0:01:38	3	0:01:33	0:01:38	1
0:01:38	0:01:41	,	0:01:38	0:01:41	1
0:01:41	0:01:42	6	0:01:41	0:01:42	1
0:01:42	0:01:44	6	0:01:42	0:01:44	5
0:01:44	0:01:50	6	0:01:44	0:01:50	5
0:01:50	0:01:52	6	0:01:50	0:01:52	5
0:01:52	0:01:56	,	0:01:52	0:01:56	5
0:01:56	0:02:03	,	0:01:56	0:02:03	5
...	...	...	...	...	...
0:04:50	0:04:54	5	0:04:50	0:04:54	5
0:04:54	0:04:58	5	0:04:54	0:04:58	4
0:04:58	0:04:59	6	0:04:58	0:04:59	4
0:04:59	0:05:03	6	0:04:59	0:05:03	3
0:05:03	0:05:04	6	0:05:03	0:05:04	2
0:05:04	0:05:07	,	0:05:04	0:05:07	2
0:05:07	0:05:09	,	0:05:07	0:05:09	2
0:05:09	0:05:13	,	0:05:09	0:05:13	1

## ► Exporting coding data

### Pooling of variables before exporting

Let's assume that you have to code the classroom behavior of 5 individual students and for that purpose you have defined 5 coding variables, one for each student. However all coding variables are defined using the same categories. Here it can make sense, with regard to statistical analyses, to export the variables not apart, but to pool (combine) their data in a single common variable. Doing so it's will be more simple to compute overall frequencies of the diverse coding categories.

To select this option, simply check the "Variables to cases" box in the [export dialog](#).

Two variables (among clip name and time intervals) are written to SPSS-File. A variable named "VALUE" contains the pooled coding values, another variable named "VARIABLE" is a string variable and identifies the name of the original variable (pointing to a individual student) to which the value standing aside belongs.

(This option requires the premium version).

## ▶ Exporting coding data

### Dividing up intervals before exporting

Using this procedure you can prepare your coding intervals for the export by dividing up it to discrete 1 sec intervals. Then a coding interval with a duration of 20 seconds will be written to the export file in the form of 20 discrete intervals each with a duration of 1 second. This can be useful for advanced computations via SPSS. An example: You intend to compare classroom activities between teachers and you would like represent the codings of each class building a graph with the x (horizontal) axis as a linear mapping of the teaching time, but you have coded using event-sampling instead of time-sampling. Therefore the length of the coding intervals is not constant but varies within and between classes. However to build a linear timescale in SPSS is feasible only if all coding intervals have equal duration and are dispersed equidistant over time. Therefore it can be necessary to dividing up the coding intervals in such a manner, that a times series arises with equal and equidistant intervals. Almost always intervals with a duration of 1 sec are needed to meet that demand

Because it is not practicable to reorganize time intervals within SPSS, videograph is providing an instrument for this purpose since version 4.1.4. Basically it is the strengthening of the option "Parallelize intervals" that can be choosed when exporting data to [SPSS](#).

Exceptionally the menu command is not located where the other commands the data export to SPSS can be found, but you can detect it in the menu "Advanced" of the timeline window under the name "dividing up intervals".

Two alternatives are available: You can divide up the data of all data bars or the data of the active data bar only. Choosing the second option (active data bar only) you can additionally choose from the coding categories, which are to be exported and which not. Therefore you are not compelled to export all codings belonging to a variable, but you can limit oneself to selected categories. That can be useful in the example above when you intend to build for each category of a variable (e.g. Lecture by the teacher vs. class work vs. silent/individual work as categories of a variable "organisation of classroom interaction") a time diagram of one's own. The categories can be selected within a dialog box that displays all categories of the active databar, and where you can click a button to select or unselect the categories for export. As soon as you have confirmed your choice by clicking the OK button, two things occur:

- The codings belonging to the categories that you have excluded are deleted from the data bars as well as from the internal data base, and only codings are left over belonging to the categories that you have picked for export.
- The time intervals for all codings leftover are splitted up to discrete intervals with duration of 1 second. When you look at the data bars thereafter, indeed no difference will be recognizable. To make visible what happened, please go to the "Options" menu of the coding window and choose under the item "Special" the command "Mark interval boundaries".

Are you choosing the first alternative (dividing up all databars) no categories can be selected, and all intervals in all data bars are splitted without further request.

When splitting or deleting codings **no undoing** is feasible. Therefore you should carry out that procedure **not** within the original project file, but use for it a copy thereof.

(This option requires the premium version).

## ► Exporting coding data

### Building tool variable for data export

Provided that certain prerequisites are met, the following applies. When exporting coding data by using the option "parallelize intervals" (see [Export-Type III](#)), you can choose in advance which range should be used for parallelizing. For this purpose there is the command "Build tool variable for data export", which you can find in the menu bar of the coding window within the pop-up menu "Options" clicking on the item "Special". Thereupon videograph generates automatically a variable named "DUMMY", which from start to finish is coded complete with a dummy-value. The length of the coding intervals has constant size, which **can be set by the user**. If this tool variable is added to the list of variables to be exported, videograph uses their coding intervals for parallelizing the intervals between the exported variables.

This export-type is applicable only for variables which are coded by means of time-sampling. Although the size of intervals between and within different variables may be different, the size of each interval must be either the same as in the tool variable or an integral multiple thereof ( e.g. 20,30,40 seconds etc., if the tool variable is coded by steps of 10 sec).

A tool variable can be useful, when different-sized intervals for export are available, but for a subsequent data analysis the intervals should be transformed into a common time scale which is divided in smaller and **equidistant** units .

## ► Exchanging data with SPSS

### Importing coding data

The commands for importing data from SPSS files can be found both in the menu in the timeline window under "Import/Export" and in the menu of the coding window under "Coding".

By importing coding data, variables of a SPSS file that have coding data can be transformed into the **data bars** of a timeline window. Please note that Videograph **does not create any new coding variables**. Instead the data bars are connected to already available coding variables. Therefore within Videograph a coding variable must be defined as being suitable to receive the imported data. The reason is that an SPSS file contains no clear information about the allowed value range of a variable.

Import is only possible when the data structure of the SPSS file satisfies certain rules of conformity. Every data bar is created on the basis of **three** SPSS variables. These variables have to be **next to each other** in a block and always have the same **sequence**.

1. The specification of the time for the **beginning** of the coding interval. The variable must be a time variable in hh:mm:ss format.
2. The specification of the time for the **end** of the coding interval. The variable must be a time variable in hh:mm:ss format.
3. **Coding values** for the intervals. The variable is numerical (F3.0) and must be a whole number type.

The time variables must show the time position in the video (starting time = 00:00:00). The names of the variables are not subject to any restrictions. **Before and after** a block there can be any number of **other** variables, yet **not** time variables (date variables, however, are allowed).

The data of each block of variables are imported as long as either all cases have been read or a system missing value is found in variables 1 or 2 (system missing values occur when there are fewer coding intervals in a variable than the number of cases in the file). A system missing value in the variables with the coding data (variable 3) is skipped as it indicates an uncoded interval.

Before importing, a relatively strict data check is carried out to avoid reading in incompatible data. To do so the following pattern is followed:

- Variable blocks with less than 3 variables are not read in.
- When one of the block variables does not have the required format (see above), the entire import procedure is terminated.

In addition, the coding data from a block will only be imported when the following pre-requisites are fulfilled:

- The values of variables 1 and 2 must be ascending. The intervals may not overlap, i.e. the value of variable 1 in case N may not be smaller than the value for variable 2 in case N-1.
- The variable with the coding (variable 3) may only contain values that belong to the value range of the videograph coding variable for which the data are destined.

The data bar created gets its name from variable 3 unless there is another data bar or coding bar within Videograph with the same name. In this case a name is derived from the videograph coding variable to which databar belongs. Value Labels and variable label of the data bar are either also imported or, if non-existent, taken from the coding variables.

Destination for the data to be imported is **always** the activated coding variable in the coding window. Data import is, thus, **only** possible when a coding variable is activated.

The import process starts with the choice of SPSS file by the user. The chosen file is then checked by videograph. The result is shown in a dialogue window, or when the file is completely unsuitable for import, an error message appears. The dialogue window is subdivided into three areas: in area 1 there are the SPSS variables that can be imported, in area 2 the names of those variables that are incompatible with the activated coding variable and can, therefore, not be imported. The user then decides which of the variables in area 1 (one, several or all) really ought to be imported by relocating the chosen elements from area 1 to area 3. There is also an information button which can be used to obtain a description of all of the shown variables.

## ■ Exchanging data with SPSS

### Hints for data import

What must be done when coding data from an SPSS file is supposed to be imported and its variable structure does not correspond to the rules (see [here](#)) that videograph stipulates for SPSS imports? The following are part of these rules:

- Every variable with coding data must immediately be preceded by two variables with the hh:mm:ss time format.
- All variables with coding data must be whole numbered.
- The file may contain any number of other variables, but none of these may have a time variable.

Should these conditions not be given, then the SPSS file has to be restructured. That is why you will find the draft of a syntax file here to automatically transform a non-videograph conform SPSS file. The SPSS commands below (macros) can then be used when not all of the variables that contain coding data are preceded by two variables specifying coding intervals, rather when, e.g., there are only two time variables for all of the coding variables. Here is a short introduction to what you can do in **such a case** (should the import not succeed for other reasons, e.g. because the variable values or the time intervals are invalid, then this solution is, of course, unsuitable):

- Following you find 2 different command blocks with SPSS-macros.
- Do mark some of this blocks complete, then copy it to the clipboard and paste it in a SPSS **syntax** window resp. syntax file (\*.SPS). The first block can be used only when these variables containing the coding data have names which are not identical up to character six. For the case that this condition did not apply the second block must be chosen.
- **Delete** all commentary lines what have been included here as explanation in italics.
- **Enter** the necessary information in the corresponding places (see below).
- Then **mark** the entire command block and **execute it**.

The following then occurs:

- All variables without coding data are deleted. The remaining variables are reformatted as double digit whole numbers (F2.0) to make sure that all of the variables have the correct format.
- For every coding variable two associated variables with time specifications are created, whereby the variable values (coding intervals) are simply taken from the two variables in the original file that the user indicated as source.
- The variables are written in a videograph conform series in a new SPSS file.

\*\*\*\*\* THE COMMAND BLOCK 1 BEGINS HERE\*\*\*\*\*.

Substitute the series `d1 d2 d3 d4 e1 e2 e3 e4 e5` with the list of the names of variables with the coding data you want to import. Should these variable names have more than six characters, the first six may not be identical.

```
DEFINE CODEVARS ()  
d1 d2 d3 d4 e1 e2 e3 e4 e5  
!ENDDDEFINE.
```

Substitute the string `'C: \SPSS \FileIn.sav'` with the complete path information from the original SPSS file with the data that is to be converted. Do not forget the quotation marks.

```
DEFINE INPATH ()  
'C: \SPSS \FileIn.sav'  
!ENDDDEFINE.
```

Substitute the string `'C: \SPSS \FileOut.sav'` with the complete path information from the original SPSS file with the data that is to be converted. Do not forget the quotation marks.

```
DEFINE OUTPATH ()  
'C: \SPSS \FileOut.sav'  
!ENDDDEFINE.
```



Substitute the word **startsek** with the name of the variable in which the beginning times of the intervals are that are to be used by videograph.

```
DEFINE VARSTART ()
startsek
!ENDDDEFINE.
```

Substitute the word **endsek** with the name of the variable in which the end times of the intervals are that are to be used by videograph.

```
DEFINE VAREND ()
endsek
!ENDDDEFINE.
```

```
*****BEWARE: CHANGE NOTHING FROM HERE ON*****
*****JUST MARK AND EXECUTE*****
```

```
DEFINE !INIT ( !POSITIONAL !TOKENS(1)
!/POSITIONAL !TOKENS(1)).
GET FILE INPATH.
FORMATS !1 (TIME8).
FORMATS !2 (TIME8).
!ENDDDEFINE.
```

```
DEFINE !MKVAR(!POSITIONAL !TOKENS(1)
!/POSITIONAL !TOKENS(1)
/!LIST = !CHAREND('/')).
!DO !VAR !IN (!LIST).
!IF (!LENGTH(!VAR) >6) !THEN.
COMPUTE !concat(!SUBSTRING(!VAR,1,6), '_S') = !1.
COMPUTE !concat(!SUBSTRING(!VAR,1,6), '_E') = !2.
FORMATS !concat(!SUBSTRING(!VAR,1,6), '_S') (TIME8).
FORMATS !concat(!SUBSTRING(!VAR,1,6), '_E') (TIME8).
!ELSE.
COMPUTE !concat(!SUBSTRING(!VAR,1), '_S') = !1.
COMPUTE !concat(!SUBSTRING(!VAR,1), '_E') = !2.
FORMATS !concat(!SUBSTRING(!VAR,1), '_S') (TIME8).
FORMATS !concat(!SUBSTRING(!VAR,1), '_E') (TIME8).
!IFEND.
FORMATS !VAR (F2.0).
!DOEND.
!ENDDDEFINE.
```

```
DEFINE !SAVE( NAME = !TOKENS(1)
/!LIST = !CHAREND('/') ).
!LET !STRING = ''.
!DO !VAR !IN (!LIST).
!IF (!LENGTH(!VAR) >6) !THEN.
!LET !STRING = !CONCAT(!STRING, !concat(!SUBSTRING(!VAR,1,6), '_S') ).
!LET !STRING = !CONCAT(!STRING, ' ').
!LET !STRING = !CONCAT(!STRING, !concat(!SUBSTRING(!VAR,1,6), '_E')).
!LET !STRING = !CONCAT(!STRING, ' ').
!LET !STRING = !CONCAT(!STRING, !VAR).
!LET !STRING = !CONCAT(!STRING, ' ').
!ELSE.
!LET !STRING = !CONCAT(!STRING, !concat(!SUBSTRING(!VAR,1), '_S') ).
!LET !STRING = !CONCAT(!STRING, ' ').
!LET !STRING = !CONCAT(!STRING, !concat(!SUBSTRING(!VAR,1), '_E')).
!LET !STRING = !CONCAT(!STRING, ' ').
!LET !STRING = !CONCAT(!STRING, !VAR).
!LET !STRING = !CONCAT(!STRING, ' ').
```

```
!IFEND.  
!DOEND.  
SAVE OUTFILE= !NAME  
/KEEP !STRING  
/COMPRESSED.  
!ENDDFINE.
```

```
!INIT VARSTART VAREND.  
EXECUTE.  
!MKVAR VARSTART VAREND  
LIST = CODEVARS / .  
EXECUTE.  
!SAVE NAME = OUTPATH  
LIST=CODEVARS / .  
EXECUTE.
```

\*\*\*\*\*THE COMMAND BLOCK 1 ENDS HERE\*\*\*\*\*.

\*\*\*\*\*THE COMMAND BLOCK 2 BEGINS HERE\*\*\*\*\*.

Substitute the series `d1 d2 d3 d4 e1 e2 e3 e4 e5` with the list of the names of variables with the coding data you want to import. Should these variable names have more than six characters, the first six may not be identical.

```
DEFINE CODEVARS ()  
d1 d2 d3 d4 e1 e2 e3 e4 e5  
!ENDDFINE.
```

Here is the chain of characters with the running index from which the names of the variables for the starting and ending times of the coding intervals can be formed. There are entries for up to 50 coding variables already given. This means you only have to make a change when you want to import more than 50 variables. In this case you have to complement the list by additional indices, e.g. .... 50\_51\_52\_53\_. Make sure the indices are correctly written: connect them with an underlining, not a space, the last character in the list must always be an underlining.

```
DEFINE INDIZES ()  
1_2_3_4_5_6_7_8_9_10_11_12_13_14_15_16_17_18_19_20_21_22_23_24_25_26_27_28_29_30_31_32_33_34  
35_36_37_38_39_40_41_42_43_44_45_46_47_48_49_50_  
!ENDDFINE.
```

Substitute the string `C:\SPSS\fileIn.sav` with the complete path information from the original SPSS file with the data that is to be converted. Do not forget the quotation marks.

```
DEFINE INPATH ()  
'C:\SPSS\fileIn.sav'  
!ENDDFINE.
```

Substitute the string `C:\SPSS\fileOut.sav` with the complete path information from the original SPSS file with the data that is to be converted. Do not forget the quotation marks.

```
DEFINE OUTPATH ()  
'C:\SPSS\fileOut.sav'  
!ENDDFINE.
```

Substitute the word `startsek` with the name of the variable in which the beginning times of the intervals are that are to be used by videograph.

```
DEFINE VARSTART ()  
startsek  
!ENDDFINE.
```

Substitute the word `endsek` with the name of the variable in which the end times of the intervals are that are to be used by videograph.

```
DEFINE VAREND ()  
endsek
```

!ENDDDEFINE.

\*\*\*\*\*BEWARE: CHANGE NOTHING FROM HERE ON\*\*\*\*\*.

\*\*\*\*\*JUST MARK AND EXECUTE\*\*\*\*\*.

```
DEFINE !INIT ( !POSITIONAL !TOKENS(1)
/!POSITIONAL !TOKENS(1)).
GET FILE INPATH.
FORMATS !1 (TIME8).
FORMATS !2 (TIME8).
!ENDDDEFINE.
```

```
DEFINE !MKVAR(!POSITIONAL !TOKENS(1)
/!POSITIONAL !TOKENS(1)
/!POSITIONAL !CHAREND('/')
/!LIST = !CHAREND('/')).
!LET !IDX = !3.
!DO !VAR !IN (!LIST).
COMPUTE !concat('S_', !SUBSTRING(!IDX, 1, !INDEX(!IDX, '_'))) = !1.
COMPUTE !concat('E_', !SUBSTRING(!IDX, 1, !INDEX(!IDX, '_'))) = !2.
FORMATS !concat('S_', !SUBSTRING(!IDX, 1, !INDEX(!IDX, '_'))) (TIME8).
FORMATS !concat('E_', !SUBSTRING(!IDX, 1, !INDEX(!IDX, '_'))) (TIME8).
FORMATS !VAR (F2.0).
!LET !IDX = !SUBSTRING(!IDX, !INDEX(!IDX, '_') ).
!LET !IDX = !SUBSTRING(!IDX, 2 ).
!DOEND.
!ENDDDEFINE.
```

```
DEFINE !SAVE(!POSITIONAL !TOKENS(1)
/!POSITIONAL !CHAREND('/')
/!LIST = !CHAREND('/') ).
!LET !STRING = ' '.
!LET !IDX = !2.
!DO !VAR !IN (!LIST).
!LET !STRING = !CONCAT(!STRING, !concat('S_', !SUBSTRING(!IDX, 1, !INDEX(!IDX, '_')))).
!LET !STRING = !CONCAT(!STRING, ' ').
!LET !STRING = !CONCAT(!STRING, !concat('E_', !SUBSTRING(!IDX, 1, !INDEX(!IDX, '_')))).
!LET !STRING = !CONCAT(!STRING, ' ').
!LET !STRING = !CONCAT(!STRING, !VAR).
!LET !STRING = !CONCAT(!STRING, ' ').
!LET !IDX = !SUBSTRING(!IDX, !INDEX(!IDX, '_') ).
!LET !IDX = !SUBSTRING(!IDX, 2 ).
!DOEND.
SAVE OUTFILE= !1
/KEEP !STRING
/COMPRESSED.
!ENDDDEFINE.
```

```
!INIT VARSTART VAREND.
EXECUTE.
!MKVAR VARSTART VAREND INDIZES /
LIST = CODEVARS / .
EXECUTE.
!SAVE OUTPATH INDIZES /
LIST=CODEVARS / .
EXECUTE.
```

\*\*\*\*\*THE COMMAND BLOCK 2 ENDS HERE\*\*\*\*\*.

## ► Help on export-dialog

The **left** window has a list of the variables that **can be** exported. In the **right** window there are the names of the variables that **are to be** exported.


Using the mouse **choose** the variables from the **left** list of variables that you want to export.

 Click here to pull the chosen variables from the **left** to the **right** window.

You can also **reverse** the procedure. Choose the variables from the **right** window that you want to **remove** from the export list and

 Click here to pull the chosen variables from the **right** to the **left** window

Normally every variable is exported **together** with its **coding interval**. The symbol for this is a [ ] after the name of the variable in the export list in the **right** window. It is, however, possible to just export **only** variable values (without coding intervals). When, for example, the coding intervals between N exported variables are identical, then the variable values can be exported alone without any loss of information for N-1 variables.

 When you click here, the symbol [ ] will be removed from all of the variables selected in the **right** window when it is there and added when it is not.

 Select a variable and click here for a description of the variable.

### Save uncoded intervals too

Time intervals with **no variable value** (coding gaps) are also exported. The SYSMISS value is inserted as the variable value for SPSS exports. The data always form a complete time series with this option without having to define a residual category for this purpose.

### Use selected play area

Before export, time intervals with a missing variable value (coding gaps) are **divided** into **partial** intervals - when possible - that are just as large as the active playback interval. The effect is that the codings form a complete time series that is parallel between the variables. This is only possible when a constant time interval has been used the entire time when coding.

### Parallelize intervals

This option is for data generated **without** a constant time interval so that incongruent coding intervals are created between the variables. When the results are transferred to an SPSS file without being transformed, time series result that may not be the same between the variables. This means the data structure is not especially suited for multivariate analysis. On activating the option, the entire coding period (i.e. both the coded and the uncoded passages) is so divided before export that identical time intervals are created for all of the variables. This means there are only identical time series in the export files, but with intervals of varying lengths within a time series. The shortest interval can be 1 second long at the most.

### Variables to cases

Let's assume that you have to code the classroom behavior of 5 individual students and for that purpose you have defined 5 coding variables, one for each student. However all coding variables are defined using the same categories. Here it can make sense, with regard to statistical analyses, to export the variable not apart, but to pool (combine) their data in a single common variable. Doing so it's will be more simple to compute overall frequencies of the diverse coding categories. (This option requires the premium version).

## ▶ Help on import dialogue

The **upper left window** contains the list of variables that **can be imported**. In the **right window** there are the names of the variables that are to be **imported**. The lower left window shows variables that cannot be imported because they have the wrong format. These are usually variables which have one or more values outside the defined value range of coding variable for which the data are destined.

In the **left** list of variables use the mouse to select the variables you want to import



Click here to drag the chosen variables from the **left** window to the **right** one.

You can also **reverse** the procedure. Select the variables you want to **remove** from the import list in the **right window** and



Click here to move the chosen variables from the **right** window to the **left** one.



Choose a variable and click here to get a description of the variable. Once the info window is opened, you can continually recall information about other variables without having to close and reopen the window beforehand.

## ► Exporting data as text

### Basic rules

In addition to SPSS export, Videograph offers the opportunity to transfer the coded data in a file in a plain text format. This allows users who do not work with the SPSS statistics program to use other programs - e.g. Excel - in order to use the coding results to calculate or to process them as graphs. As an additional option the data can also be copied into the Windows **clipboard**. From here they can be inserted into other programs ( Excel, SPSS, Word) by using the "paste" command, e.g. to create new variables in a SPSS data window or to insert a column in an Excel table.

The menu commands for exporting coding data in a text file can be found both in the timeline window menu under "**Import/Export->export as text**" and in the coding window menu under "**coding->data export tab-delimited**". When only one video clip has been opened it makes no difference which menu is used for the export. In the other case the following holds true: coding data can only be exported from the active time line window using the export command in the time line window; with the same command in the coding window menu the coding data from all of the existing timeline windows can be exported.

When the export command is accessed in one of the two menus, a dialog window opens. Here the data bars that are to be exported can be selected from a list. It is also possible to choose between various export **options** which will be explained later. A file name must now be entered and the export can then follow. The file where the coding data are is a ASCII text file with a structure that can be used universally. It is made up of columns and rows, between the columns there are tab markers, at the end of every row there is a carriage return. Such a format is also called "tab-delimited" and its advantage is that e.g. spreadsheet programs like **Excel** recognize it as a **table** and import it as such direct. In the case of word processing programs the case is similar. Here there is usually a "text in table" command (or something similar) with which every tab-delimited text can be brought into a table form. A data export in 2 other formats is also possible, see below for details.

The first column in the export file is the **name** of the clip including its path. Then come the variables with the coding data. Here there are **two alternatives** for each data bar:

- The coding data are arranged in **three** columns, one after the other:
  - a) Column 1 contains the times for the **beginning** of the coding intervals.
  - b) Column 2 contains the times for the **end** of the coding intervals.
  - c) The **coding values** are in column 3.
- The times are for the time position in the video (starting time = 00:00:00).
- or:

There is only **one** column with the coding values and the coding intervals are not transferred. This procedure makes sense when some of the data bars have identical coding intervals.

The data in each data bar are written as cases in the same sequence in the SPSS file in which they follow each other in the data bar, i.e. the variables 1 (interval begin) and 2 (interval end) form a time series sorted in ascending order. Empty data bars that do not yet contain any data will not be exported.

Beyond the already mentioned choices the dialog window offers three other important options (in the form of checkboxes) for exporting. Two of them have a decisive effect on the data structure of the export file, the other determines the destination, as:

- it gives the user the choice of whether the data is to be exported into a file **or** (in the same format) to the Windows **clipboard**. When the coding data are sent to the clipboard, other programs ( e.g. **Excel**, SPSS, Word) can access the data by means of the "paste" command. New columns can thus be added ad-hoc to an Excel table or new variables in an SPSS data window can be created.
- The first of the two options which can influence the data structure of the export file is characterized with "**save uncoded intervals to**". This means that the time intervals **without variable value** (coding gaps) are also exported. The result is that all time columns form a complete time series without having to define a residual category for this purpose.
- The other option is the "**use selected play area**" one. All time intervals with coding gaps are divided - as far as possible - into **partial** intervals that are just as long as the active playback interval. The effect is that the codings form a time series which is not only complete but also is identical for all variables. This is only possible when a constant time interval has been used the entire time when coding.
- A third option, "**parallelize intervals**", is an alternative option to the two previous ones. It is for data that was created **without** or **without a constant** time interval so that incongruent coding intervals occur between the

variables. When the results are transferred to an file without being transformed, time series result that may not be the same between the variables. This means the data structure is not especially suited for multivariate analysis. When choosing the option "parallelize intervals", the entire coding period (i.e. both the coded as well as the uncoded passages) are so subdivided before export that all variables have identical time intervals. The export file, thus, only has identical time series but with intervals of varying length within a time series, whereby the shortest interval can be 1 second long.

More concrete information about this topic may be found in the sections "[Export Type I](#)", "[Export Type II](#)" and "[Export Type III](#)" in the chapter on SPSS Export where some prototype data structures are presented and explained. The explanations given there also hold true for data to be exported in text format.

A **further option** for export of coding data into a text file is - when there is a transcribed text - to append this to the exported codings as the last column. This functions according to the following principle: The texts for the transcribed time interval are transferred **line for line** chronologically **without indication of time** in the export file. Therefore it only makes sense to use this option when the **coding and transcription intervals are exactly the same** so that there is a clear correlation between the lines of the table and the time interval belonging to it. Should this not be the case, then a valid temporal assign of transcription text is not possible either.

(Should you desire a joint export of transcribed text and transcription intervals, then the transcript must be exported as a separate file.)

## Data export to other file formats

By default, Videograph offers the file format "Text file, tab-separated (\*.txt;\*.dat)" in the file dialog for export to choose from. This is the text file with a table structure mentioned at the beginning.

Instead, you can choose between 2 other formats.

One has the same tabular structure as the tab-separated format, but does not contain any tab characters, but instead semicolons. And the data fields are framed by single quotes. To do this, click on the selection box for the file type and select the option "Text file CSV format (\*.csv)". A CSV file is the ideal format for reading the coding data into the statistical software R as a table in order to perform statistical calculations. In general, the CSV format is a very universally used format.

As a third alternative, you can use the "Text file SQL-Query-Format (\*.sql)" option as the file type. You will then receive a text file in which the coding data are embedded in SQL commands so that a data table for a MySQL database can be created from it.

## ▶ Help on export-dialog

The **left** window has a list of the variables that **can be** exported. In the **right** window there are the names of the variables that **are to be** exported.

Using the mouse **choose** the variables from the **left** list of variables that you want to export.



Click here to pull the chosen variables from the **left** to the **right** window.

You can also **reverse** the procedure. Choose the variables from the **right** window that you want to **remove** from the export list and



Click here to pull the chosen variables from the **right** to the **left** window

Normally every variable is exported **together** with its **coding interval**. The symbol for this is a [ ] after the name of the variable in the export list in the **right** window. It is, however, possible to just export **only** variable values (without coding intervals). When, for example, the coding intervals between N exported variables are identical, then the variable values can be exported alone without any loss of information for N-1 variables.



When you click here, the symbol [ ] will be removed from all of the variables selected in the **right** window when it is there and added when it is not.



Select a variable and click here for a description of the variable.

### Save uncoded intervals too

Time intervals with **no variable value** (coding gaps) are also exported. The SYSMISS value is inserted as the variable value for SPSS exports. The data always form a complete time series with this option without having to define a residual category for this purpose.

### Use selected play area

Before export, time intervals with a missing variable value (coding gaps) are **divided** into **partial** intervals - when possible - that are just as large as the active playback interval. The effect is that the codings form a complete time series that is parallel between the variables. This is only possible when a constant time interval has been used the entire time when coding.

### Parallelize intervals

This option is for data generated **without** a constant time interval so that incongruent coding intervals are created between the variables. When the results are transferred to an SPSS file without being transformed, time series result that may not be the same between the variables. This means the data structure is not especially suited for multivariate analysis.

On activating the option, the entire coding period (i.e. both the coded and the uncoded passages) is so divided before export that identical time intervals are created for all of the variables. This means there are only identical time series in the export file, but with intervals of varying lengths within a time series. The shortest interval can be 1 second long at the most.

### Export to clipboard only

The data are not written in a file, instead they are transferred to the Windows clipboard. Here they can be taken over by another program (e.g. Excel, SPSS, Word) via the "insert" command and used, e.g. to generate new variables in an SPSS data sheet or to add columns to an Excel spreadsheet.

### Include transcript

When there is a transcribed text it can be appended to the exported codings as the last column with this option. This functions according to the following principle: The texts from the transcribed time intervals are transferred **line for line** in chronological order, but **without** indication of times, to the export file. Therefore it only makes sense to use this option when the coding and transcription intervals are absolutely **identical** so that there is a clear relationship between a line in the table and the time interval that belongs to it. Should this not be the case, then it is not possible to set the codings and transcript text in a correct temporal relation.

Should you wish to export transcribed text together with its time indications then you have to export the transcript as a separate file.



## ► Exportings transcripts

### Exporting to an SPSS file

The commands for exporting transcripts to SPSS can be found both in the menu in the timeline window under "Import/Export" and in the menu of the transcript window under "transcript". The name of the SPSS file can be stipulated in the dialogue window, but make sure the file ends with **.SAV** when choosing the name. No data can be added to existing files (due to the restrictions within SPSS); should there be a file with the same name it will be written over.

When only 1 videoclip is loaded, it makes not difference which menu is exported from. Otherwise the following holds: Only the activated transcript is exported with the export command in the transcript window menu. The same command in the timeline window menu exports the transcripts of all of the loaded clips.

When exporting, the transcript data is written in 4 successive SPSS variables which are in the following sequence.

- Name of the clip including path. The variable is a STRING type and can take up to 255 characters.
- Specification of time for the beginning of the transcribed time interval. The variable has time format in hh:mm:ss style.
- Specification of time for the end of the transcribed time interval. The variable has type format in hh:mm:ss style.
- Variable with the transcript text. The variable has the same format (A255) as variable 1. When the text exceeds the maximum length of 255 characters additional string variables (A255) are appended as long as the whole text has been placed. When there are notes, they are appended (up to 255 characters) as own variable that's name begins with the string "COMMENT".

The specifications of time describe the time position in the video (Starting time = 00:00:00). When forming the variable names the following procedure is followed:

- The name of variable 1 (clip name) begins with the prefix TS, followed by the number of the video and the supplement FILE, e.g. TS2FILE, which means that it is the transcript for video 2.
- The names of variable 2, 3 and 4 follow this scheme, but with another appendix, viz. START (1), END (2) und TXT (3).

When transcribing interval of greater length so that the text must be splitted in several variables it is suggested to put the export in a **text file**- an option that videograph also offers. In this case there is no limit to the length and the transcript is easier to read and survey as continuous text.

## ► Exporting transcripts

### Exporting as ASCII-text

The commands for exporting transcripts to text files can be found both in the menu in the timeline window under "Import/Export" and in the menu of the transcription window under "transcript". The name of the text file can be set down in a dialogue window. The transcript of the clip that's timeline window is activated is always the one to be exported. When there are notes they are also exported and appended to the transcript text. It is also possible to choose whether the text **and** the transcription interval or, if no specification of time is needed (e.g. in the transcript of an interview), only the text is to be exported. In the last case the "**export without time intervals**" checkbox must be clicked on.

When exporting, videograph adds additional tabs and carriage returns to the text to make it clearer. The best presentation results when the export file is read in Microsoft **Word** (or another program that has a table option) and then transformed into a **table**. When transformed, the tabs become columns in the table which creates an optimal picture. Depending on preference, the table may be framed or unframed.

Additional tabs inside of transcription text can be used to visualize partitions. For example when within the same time interval several people are speaking, you can via tabs place the statement of every person in a own column, by what it is easier to identify the sequence of speakers. Not recommendable yet is a unsystematic use of tabs because it can produce a unclearly arranged view. The same is true for tabs within the notes. When there are tabs in the notes, they are deleted before the export by Videograph and replaced with spaces.

Also exporting to **Excel** (or another similar program) functions without problem as long as the corresponding option (tab-delimited text) in the opening file dialogue of Excel is chosen. Excel reads the transcript and transforms it into a spreadsheet.

Using the checkbox "**optimize for table format**" it is possible to chose between two slightly different layout variations. The standard variant (without activated checkbox) are better suited for use in a normal text editor like e.g. Notepad, the variant with the checkbox on is for importing to Word or Excel with following transformation into a table. The effect is that the time indications are put in the same line of the table as the transcript text which depending on the number of intervals can clearly reduce the size of the table. When the option is switched off, the time indications are in a separate headline which improves legibility should the text not be read as a table, but as a continuous text.

To make things clearer, the following example shows the view of a transcript formatted as a table (without frame) in Microsoft Word:

00:05:40 - 00:05:50

T Where does electricity play a role

00:05:50 - 00:06:00

T normally?

S2 I think in everyday life. Almost everywhere.

T Where in this room? Let' stay in this room.

00:06:00 - 00:06:10

T This is now everyday life. Let' stay in this room.

S20 Yes, for example light.

T The light, right.

S2 For the cameras. (I think).

T The cameras need electricity. For what else?

00:06:10 - 00:06:20

S25 In the home. For the household.

T Where in the home? Where exactly?

S25 For kitchen appliances or

T kitchen appliances.

S2 Heating.

T yes.

00:06:20 - 00:06:30

T What makes a kitchen appliance work? What makes a light bulb burn? Ah, what do you believe you need for it?

00:06:30 - 00:06:40

T S10.

S10 Electricity.

T Electricity. Mh. Where does it come from?

S? {whispering} From the outlet.

T Pardon.

S7 From the

00:06:40 - 00:06:50

S7 power plant.

T You believe from the power plant. (pardon?)

## ► Exchanging data with SPSS

### Importing transcripts

The commands for importing transcripts from an SPSS file can be found both in the timeline menu under "Import/Export" and in the transcript window menu under "Transcript".

Using the transcript import option, a video can be assigned a synchronized accompanying text, found in a SPSS file as a string variable. Text can mean a transcript in the strict sense of the work as well as any other text material. One conceivable use would be the import of transcripts that were created with another program and are now to be used within videograph. Should the file to be imported not have that data structure videograph requires, there are SPSS syntax commands available which usually make converting easy.

Even when several clips are loaded, importing must be done for each clip individually. The **destinated clip** when importing is always the one whose name is shown in the title bar of the transcript window.

Import is only possible when the data structure of the SPSS file conforms to certain rules. Every transcription interval is created on the information basis of **three** SPSS variables. These variables have to always be **next to each other** in the block and in the same **sequence**.

1. The specification of the time for the **beginning** of the transcribed interval. The variable is a time variable in hh:mm:ss format.
2. The specification of the time for the **end** of the transcribed interval. The variable is a time variable in hh:mm:ss format.
3. Variable or variables containing the transcript text. Only variables formatted as string A255 which can take up to 255 characters are valid. The transcript text may be divided into several variables. This is always necessary when it is longer than 255 characters. Should this be the case, then the variables always have to be **next to each other** and so sequenced that every following variable also contains the following text for the previous variables.

The time variables must show the time position in the video (starting time = 00:00:00). The names of the variables are not subject to any restrictions. **Before** and **after** a block there can be any number of **other** variables, yet **not** time variables (date variables, however, are allowed).

The data from a variable block are read in until all cases have been read or a system missing value is found. In contrast to the import of coding data the reading process is abruptly as well when a system missing value is found in variable 3 (first variable with transcript text), i.e. all of the time intervals listed must contain transcript text. Before importing, a relatively strict data check is executed to avoid reading in incompatible data. To do so the following pattern is followed:

- Variable blocks with less than three variables are ignored.
- When one of the block variables does not have the requisite format (see above), the entire import process is broken off.

In addition, the data from a block are only imported when the following pre-requisites are fulfilled:

- The values of variables 1 and 2 must be ascending. The intervals may not overlap, i.e. the value of variable 1 in case N may not be smaller than the value for variable 2 in case N-1.

The text for a transcript interval is taken from or constructed from the data cells of **all** string variables that a) directly follow the variables 1 and 2, b) are in a continuous order **and** c) are in the (A255) format. i.e. text read in from consecutive string variables is linked as long as Videograph meets a variable not formatted as string A255.

When a variable contains no transcript text but associated notes, its name must begin with "**COMMENT**"; the result is that the data are not shown in the transcription window but only in the note window. Only 1 variable containing notes (up to 255 characters) can exist.

When your SPSS file has string variables in A255 format that contain **no** transcript text or notes, but rather e.g. the **pathname** for the video clip, make sure there is a spatial separation or insert a dummy variable of a different type or of another format.

At the beginning of the procedure the user chooses an SPSS file. The chosen file is then checked by videograph. The result is shown in a dialogue window, or when the file is completely unsuitable for import, an error message appears. The dialogue window is subdivided into three areas: in area 1 there are the transcripts that can be imported, in area 2 the variable names of those transcripts that cannot be imported. The user then decides which transcript in area 1 really ought to be imported by relocating the chosen element from area 1 to area 3.

## ► Exchanging data with Videograph

### Importing transcripts

You can read in transcripts not only from a SPSS-file but also from another Videograph-file (\*.VDG). This option may be useful when you want to read in or to correct transcripts, but the original clip is not available at the time.

The clip used to read in should not be shorter than the original clip because otherwise the transcription text is readed in not completely.

You have the choice between four options: the transcription text without notes can be imported, or the notes without transcription text, or either together. In addition the imported text can be appended to already existing text or existing text can be overwritten with it. In the appending-mode, no time intervals are imported, but solely text, and this only if it belongs to an interval for that already exists a transcription interval with identical start time.

The command for importing a transscript from other videograph files is located in the transcription window menu under the menu item 'Transcript->Import'.

## ► Exchanging data with ASCII-Files

### Importing transcripts

Since version 4.2.1 transcripts can be imported from a normal ASCII-text file, provided that the file is builded conform to specific but uncomplicated rules.

- The transcript text in the file must be partitioned into blocs, each one must be preceded by a line with the declaration of the time interval to which the bloc belongs. The declaration has to stand in a separate line, it must be enframed by squared brackets, it must enclose start time and end time of the interval and has to look as follows: [h:mm:ss-h:mm:ss], i.e. [0:00:30-0:01:25]. Other alphanumeric characters (with the exception of spaces) in the same line are prohibited.
- Subsequently the text belonging to the interval is standing. It ends by definition where the next line with the declaration of a time interval begins. There are no restrictions regarding the composition of the text, it can contain any characters that is printable, solely leading and trailing blank lines are removed by Videograph.

Here is a short excerpt from a text file suited for importing to videograph as a transcript:

[00:05:40 - 00:05:50]

T Where does electricity play a role

[00:05:50 - 00:06:00]

T normally?

S2 I think in everyday life. Almost everywhere.

T Where in this room? Let' stay in this room.

[00:06:00 - 00:06:10]

T This is now everyday life. Let' stay in this room.

S20 Yes, for example light.

T The light, right.

S2 For the cameras. (I think).

T The cameras need electricity. For what else?

115

[00:06:10 - 00:06:20]

S25 In the home. For the household.

T Where in the home? Where exactly?

S25 For kitchen appliances or

T kitchen appliances.

S2 Heating.


T yes.



The menu command for importing a transcript from a ASCII-textfile can be found on menu "transcript" of the transcript window under the item "Import.../Ascii-file".


Having the import file readed in, its content is showed in a text window. In addition the time intervals found in the file by videograph are listed separately in a list box. By clicking on any element of the list you can jump to the line in the text window where the text belonging to the interval is standing. Now you can verify the complete window text, including all interval declarations. Corrections you can write directly to the fully editable text window. Videograph will give the permission for importing then, and not till then, when following conditions apply:

- It must exist at least 1 correct formatted declaration of a time interval
- The declarations of the time intervals must be serialized in ascending order, no start time may be lower as the end time of the preceding interval and it must be higher as the start time of the preceding interval. Intervals that are violating that rule can be identified in the list box by means of 2 appended question marks (??).

Videograph cannot recognize whether or where intervals are missing. Likewise he is unable to identify declarations of time intervals that have an erroneous format. In that case he will assume that the erroneous declaration line is a part of the transcript text. Hence it is important to edit the whole text exhaustive before the command for importing is given

However Videograph can give feedback, whether and where there is a time gap between the start time of a interval and the end time of the preceding interval. Using the command "mark unfilled intervals" in the menu bar videograph marks such intervals by inserting the character <. Alternatively you can click on the button  on the toolbar.

After modifying the content of the text window ( by correcting/inserting declarations of time intervals or  by eliminating spelling mistakes), videograph will take over the modifications then, and not till then, when you have pressed the button  on the toolbar.

When you believe that all errors are cleared, and videograph agrees, and when you want it's time to start the import, then press the button  top right on the toolbar

No method exist to distinguish or mark notes or comments assigned to the transcript text. Videograph will not recognize such elements and consider it as a integral part of the transcript text

(This option requires the premium version).

## ▶ Playlists

### Putting together a playlist

When you have the extended version of Videograph with the supplementary module (can be delivered separately) you can compile user-defined passages from the video (or audio) in a playlist and then play them selectively. Playlists can be stored independently of the project file to be imported later so that they are available universally. There are 3 variants to choose from when compiling a playlist

- You can filter out all passages which correspond to a certain criterion, e.g. you can decide that all of the passages in which the teacher reacts to a student's question in the playlist are to be included.
- You can select playback intervals in the timeline window
- Variations 1 and 2 can be combined.

#### Determining the filtering conditions

Select the selective replay option under Advanced in the timeline menu. In the upper left window of the dialogue box you will find the names of the coding variables. Only those variables are shown for which there is a coding. Highlight the name of a variable and in window to the right of it a coding category. Then push the button ▼. Your choice will then be shown as the filtering condition in the two lower windows of the dialogue box. Repeat the procedure with other coding variables when you want to add further criteria. In this case you may choose between the two logical operators AND or OR. AND is the default choice. To use OR click on the checkbox. When more than 2 coding variables are chosen for a filter it is not possible to mix both operators. An alternative possibility is to split the desired condition into two or more filters and then adding these filters one by one to the playlist. How to do this is shown [below](#). The same procedure is necessary to create an OR connection among several categories of the same coding variable as only one coding category for each variable can be chosen per filter.

When the filter conditions have been completely defined, press the button named Show. A new window with a playlist will open which shows in chronological order the video time intervals that fulfill the conditions defined in the filter.

#### Choosing playback intervals

As above choose the selective replay option in the menu advanced of the timeline window. Instead of marking the name of a variable, click the Show button immediately. The playlist window then appears, yet without a playlist (the error message "no data" first appears, but it can be ignored). Now activate the timeline window, select a playback interval using the two area markers of the time ruler and **activate** it. Now click the Add button and the interval will be added to the playlist. Before doing so you can provide the chosen time interval with a commentary (up to 255 characters). In order to add additional time intervals, proceed as above.

#### Combining variants

Proceed exactly as described in determining the filtering conditions. After the window with the playlist has been opened you can add as many other intervals as desired to the existing playlist by following the method described under Choosing playback intervals.

#### Loading and storing playlists

After you have builded a playlist, **you should store it immediately** since it is otherwise erased when the list window is closed. In order to store, push the Save button and choose a file name. It is possible to store in both dialogue windows. In the filter window the filter conditions are stored (file extension \*.KRL), while in the playlist window the resulting time intervals are stored (file extension \*.PLL). The stored files can be re-opened using the Load button. The filter window is used to open a stored KRL-file and the playlist window for a PLL-file.

#### Revising lists

You can expand a stored playlist (\*.PLL) anytime by opening it in the playlist window and then adding and storing further intervals. You can add both additional filters as well as choose new playback intervals.

#### Adding filters

Select the option selective replay found under advanced in the menu of the timeline window. In the dialogue window for the filter selection mark no variable name. Instead immediately press the Show button to open (the still empty) playlist window. Now load a stored playlist (\*.PLL file). Then choose - with the opened playlist window (**important!**) -



the selective replayoption in the advancedmenu again and add new filter conditions in the filter win dow. As soon as you push the Showbutton, the suitable intervals - should they exist - will be filtered out and added to the playlist window, where they can be run by mouse click like the other intervals.

### **Adding playback intervals**

(As before) select the selective replayoption under advancedin the timeline window menu and push the Showbutton. In the playlist window you open a stored playlist (\*.PLLfile). Add new interval to the list by selecting playback intervals from the timeline window and then pushing the Addbutton.

### **Info-notice**

By pressing the Infobutton it is possible to have a description of all the elements in the playlist. Intervals belonging to a filter indicate the filter conditions, whereby the OR operator is indicated by a| and the AND operator by a &. Intervals that have been selected direct from the timeline window are shown with the commentary given by the user.

### **Playing back the playlist**

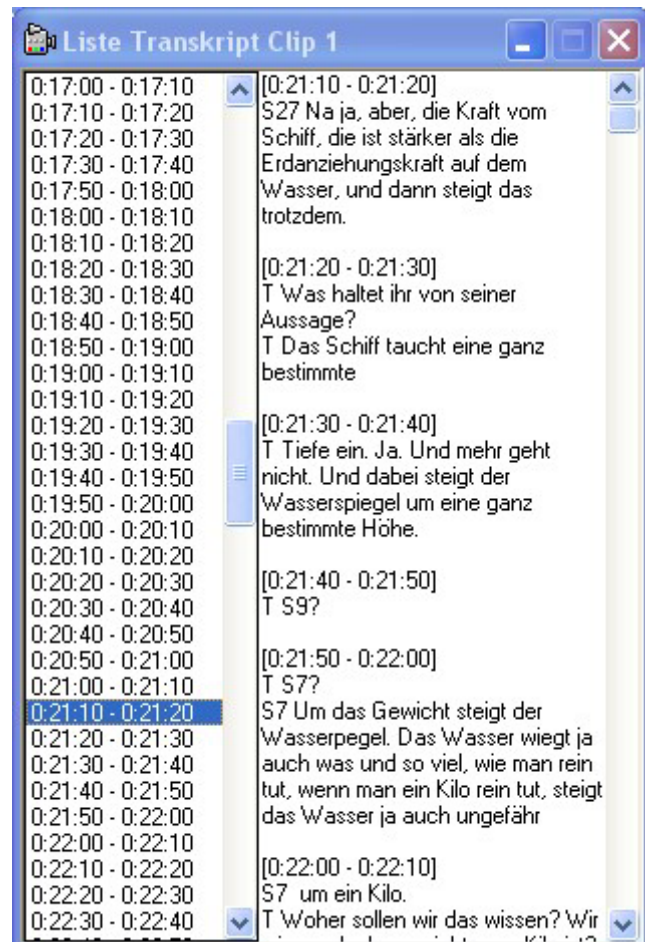
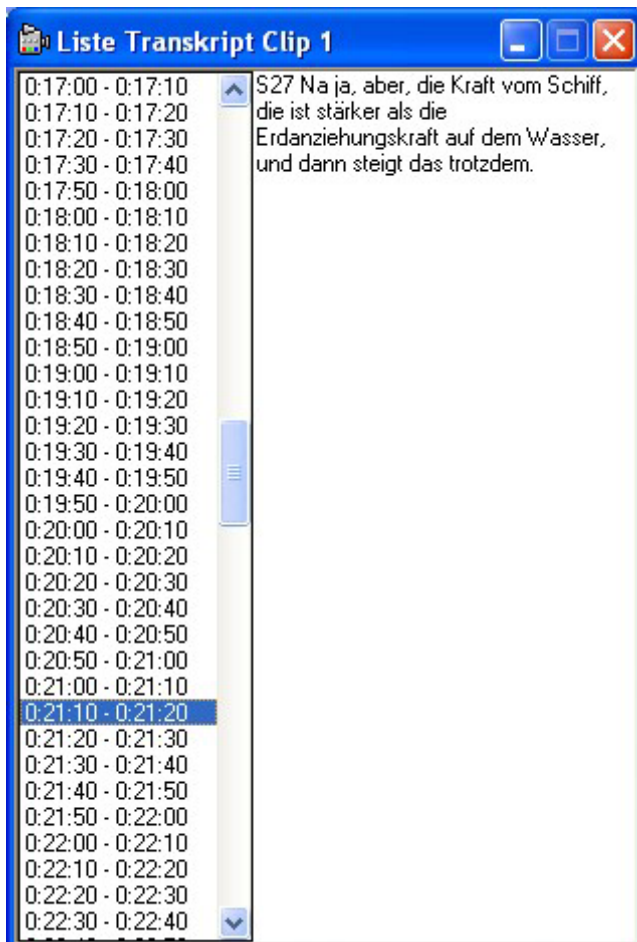
The playlist elements can be played back by marking an interval and double clicking it or by pressing the Playbutton.

## ► Transcript

### Viewing transcript in survey window

Using the the extended version of Videograph with the supplementary module (can be supplied separately), it is possible to better go through and scan the individual blocks of texts which make up the transcript as well as the transcript as a whole than is possible in the normal transcript window. A so-called overview window can be opened where a list of all of the transcribed intervals are on the left side and the transcribed text is on the right. For the most part, you can choose which section of the text is to be visible. The entire transcript can be shown, or just larger or smaller parts of it, or even just the text of an individual interval. To open the overview window, select the "open list" command under "transcript" in the transcription window. There are three variants for viewing, whereby the interval list on the left side of the window is common to all of them. With a double click of the mouse on an entry the clip moves to the beginning of the selected time interval. At the same time the playback interval is automatically directed and activated to match the transcription interval. It is possible to correct spelling mistakes in the text found in the overview window, for example, by using this method. This is much faster than having to move forward step by step using the playback interval in the timeline window. With a short double click the clip jumps to the goal so that it is possible to correct the text without having to take any other measures. The only other thing that has to be done is to turn off the writing protection.

The difference between the three variants lies in what is seen in the right hand part of the window. When an interval in the left hand list is selected by a mouse click, in the first variation only the text belonging to this interval will be shown (see picture on the left). In the second variation not only the text from the selected interval is shown, all of the following ones are also visible. When the first transcription interval is chosen the entire transcript from beginning to end can be read. This variation, however, shows no temporal references for the text. The third variant (picture on the right) is for this. The text is divided into time intervals by the added lines which subdivide the text by times.



Variant **1** always is active when the command "open list" is chosen. In order to switch on variant **2**, the command "continuous" is chosen from the "options" menu (**Beware**: The overview window has its own menu and has to be activated in order for it to be visible.). For variant **3** the switch is made using the "show intervals" command.

However the options menu has any more commands. With "mark unfilled intervals" missing intervals in the left interval list, coming from untranscribed positions in the clip, are tagged. Using the command "show notes" not the transcription text but the appropriate notes are displayed.

**In addition** the overview window has a search function in all of the variants (which can be found under "Edit" in the menu bar). Using this function the transcript can be searched for any character string. By combining the search function and steering the clip using the interval list in the left part of the window, the overview window is excellent for re-working the transcript.

The text inside of the right part of the overview window can be **printed** out as well as **saved** as text file. Using this alternative instead of applying the normal Export-command, the transcript is not formatted in table form indeed, but you have a helpful alternative solution if you need or want to have the transcript in form of a common continuous text. The commands for printing and saving you will find on the file-menu of the overview window.

## ► Coding

### Report window for Codings

In the **premium** version of videograph you can open a window presenting a report of all existing coding data and coding variables. This report window pops up when selecting the option "Open variable list" under the menu item "Coding" of the coding window. You have to activate the coding window before you can use the option.

The report lists the names, labels, values and value labels of all coding variables, as well as gives a statistical summary of the coding data existing in the timeline window. The report can be printed out as well as saved in a text file, the commands you can find on the menu of the report windows under the item "File".

The report window has two parts. The left window contains the names of the coding variables, in the right window the informations described in the following are provided:

(1)	(2)	(3)	(4)	(5)	(6)	(7)
272	174	126	0	4	4	6
0:26:12	0:05:22	0:12:00	0:00:00	0:00:09	0:00:06	0:00:12
56.98%	11.67%	26.10%	0.00%	0.33%	0.22%	0.43%

At first variable name (in brackets) and variable label are showed. Under it the defined coding values and value labels are given. It follows a table containing a summary of the coding data in the timeline window: in the first line the coded values (in parentheses) are given, followed by a data array containing the number of coded time intervals per value. The next line shows the total time span falling upon an value, and the third line displays that time span percentaged to the overall duration of clip.

If data arrays are so long that they don't fit in a single line, a line break is inserted. To avoid that you can broaden the window.

When comparing the graphic above with the picture below you can see that not only a single variable can be displayed, but also multiple variables simultaneously. At the picture above the variable selected in the left window is presented. At the picture below not only that variable is displayed, but also **all** variables **following it** (to save space in the graphic only one variable follows). To activate the lower display mode you have to choose the command "Continuous" under the menu "options" of the report window, don't forget to activate the window before.

variable list clip 1									
uphas	[Speaker]								
Speaker	Who speaks								
sozform									
1	teacher								
2	student								
3	none								
4	other								
5	teacher and student at the same time								
6	two students at the same time								
7	several students at the same time								
data:									
(1)	(2)	(3)	(4)	(5)	(6)	(7)			
272	174	126	0	4	4	6			
0:26:12	0:05:22	0:12:00	0:00:00	0:00:09	0:00:06	0:00:12			
56.98%	11.67%	26.10%	0.00%	0.33%	0.22%	0.43%			
[sozform]									
organisation of classroom interaction									
0	none								
1	lecture by the teacher								
2	dictation								
3	class discussion/class work								
4	silent/individual work								
5	working with a partner								
6	group work								
7	several kinds of work at the same time								
8	transition								
9	other								
data:									
(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
18	33	0	53	9	0	140	0	23	0
0:03:00	0:05:30	0:00:00	0:08:50	0:01:30	0:00:00	0:23:20	0:00:00	0:03:50	0:00:00
6.52%	11.96%	0.00%	19.21%	3.26%	0.00%	50.74%	0.00%	8.34%	0.00%

A few other useful features of the report window are:

- By double-clicking a variable name in the left window the associated coding bar in the coding window is scrolled to the top window position; when simultaneously the SHIFT- key is pressed, the coding bar is moved (by re-sorting) to the first position in the queue.
- In the right window you can mark off at will single or multiple data lines with the mouse, and copy it via the shortcut STRG+INSERT to the clipboard of window. Thereafter the data can be pasted to another application program, for example Excel, and used to make charts.
- The coding data displayed are based by default on the total duration of the clip. But it's possible also to determine a shorter interval, namely by the command "Select interval" in the menu "options" of the report window. After doing so three new data arrays are appended to the data arrays already existing. Please note that the percental data (third line) for interval portions are related to duration of the selected interval and not to the duration of the clip.

## ▀ Coding data

### Diagrammatical representation of coding data

Using the command "Diagram" in the shortcut pop-up menu of a databar the coding data are presented as colored bar diagram. The shortcut menu is activated by clicking the right mouse button on any point inside the head of a databar.

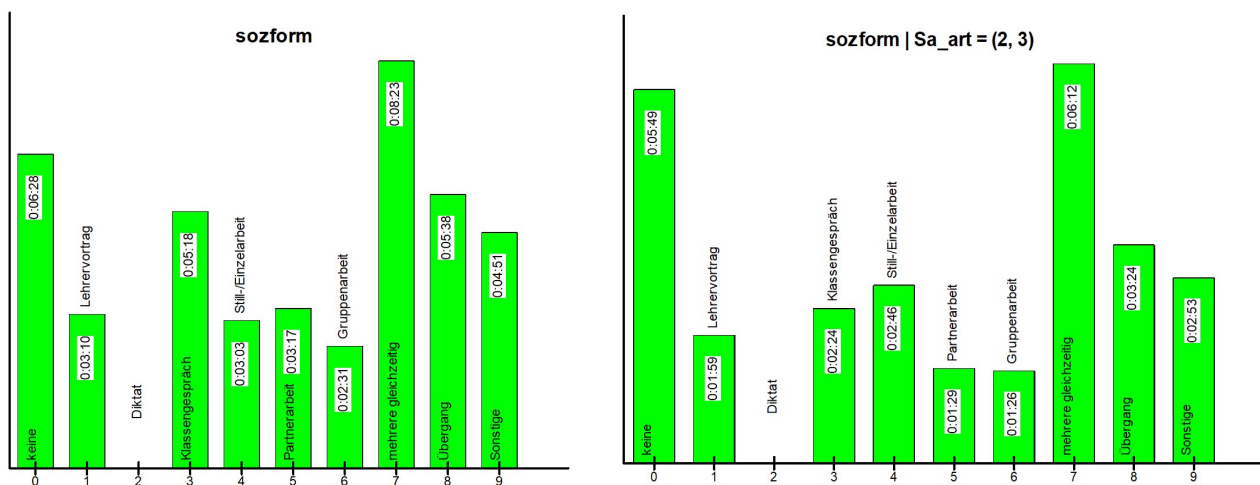
The diagram can be scaled continuously variable by changing the size of the output window. The graph can be printed out, saved as well as copied to the clipboard for use and working it in other programs. When saving or copying a format named "enhanced metafile" is employed so that the graph can be scaled without loss of quality. All commands for manipulating a diagram can be found in the menu bar of the main window, however they are only visible when the diagram window is activated. Opening the submenu "Options" you can determine if absolute or proportional time data are to be displayed.

But the coding data of a data bar can not only be output as a one-dimensional frequency distribution, but you can also display the bar chart for its conditional distribution relative to one or more coding categories of another data bar. To do this, select in the menu for the diagram window (see above) in the submenu "Options" the command "Select filter variable" ;. In the dialog window that opens, first click on the name of the data bar and then on the coding categories whose statistical relationship with the frequency distribution you are interested in. Both 1 and more categories can be selected.

The option "Select filter variable" behaves like a switch, it can be switched on and off by simply clicking with the mouse. It is initially switched off after the program has started. After you have created a window with a first bar diagram, a menu belonging to the diagram window appears in the menu bar of Videograph. where you can turn on the option. This immediately opens a new dialog window in which all data bars are listed together with their coding categories. Here, select the data bar (i.e. the filter variable) and the category or categories that you want to include as boundary conditions in the calculation. After you have made your choice and closed the window, the current bar graph is deleted and immediately redrawn. From now on, whenever you create a bar chart, you will first see the dialog box. If you do not want to set a filter variable in individual cases, simply close the window again. Otherwise you will get a bar chart with the frequency distribution of the coding categories under the condition that at the same time the filter variable has a coding in one of these categories you selected.

Deactivating the option has no effect on existing diagram windows

The diagram color looks like the color of the corresponding databar and each diagram bar is labeled with the value label of the coding category belonging to it. Both attributes can be changed any time and without to close the diagram window by simply altering the corresponding attributes of the databar ( by means of its shortcut pop-up menu) and thereafter choosing the command "diagram" once more.



The figure on the left shows the bar chart for the data bar of the coding variable **sozform** with the frequency distribution of their categories independent of the coding value of a second coding variable. The right figure demonstrates the conditional frequency distribution of the same coding variable, i.e. the distribution when the variable **sa\_art** (type of student work phase) is coded at the same time with the value 2 or 3.

You can not only illustrate the coding data totally from beginning to end, but also any more shorter periods can be selected. Create for this purpose a [playback interval](#) , in order to specify the time period which shall be showed, and activate it by double clicking a position somewhere between the two markers, so that the interval takes on a blue color.

Then the databars in the diagram will present only the data from within the time period, which the playback interval encloses.

,

## ► Coding

### Exporting databars as time diagram

For availability of this function the premium version of Videograph is required.

Using the command "export timeline as graphic" in the menu "Advanced" the databars of the timeline window in what the time flow of the codings is displayed, can be converted into a graphic and externally saved as EMF-file (enhanced meta-file). Graphics in EMF-Format are vector graphics that can be sized continuously without loss of quality and thus can be used for representations of any format, e.g. by applications as powerpoint.

In order to be flexible in the choice of the databars that are to be included in the graphic, only the currently **visible** ones are enclosed, whereby the visibility of the bars depends on the height of the timeline window and on its order. Or stated differently: the data bars that are not to be included must **completely** disappear from the visible area of the timeline window. This is not difficult to do as the databars can be resorted in any way. The data bars to be included are relocated in the front positions and/or the timeline window is moved to a height in which only the data bars that are to be enclosed are visible. Should the maximum height of the timeline window not suffice to make all of the desired elements visible, then two or more graphics are made.

The same is true for the horizontal direction. Only the databars in the **period** visible in the timeline window are displayed. It is then possible to vary the time span at will, either by scrolling horizontally or by sizing the width of the timeline window. The same effect may be reached by altering the time raster in the time ruler. The smaller the unit is, the longer the time period represented is. When you want to represent the entire stretch of time at once, you can set the time raster to "all".

Additionally the owner window of the graphic has its own menu, it appears when the window is activated. Here you will find the command "select range" whereby you can fix the period to be represented accurate to a second by entering start and end time of the desired interval. Alternatively the time span showed can be adjusted by increasing or decreasing the distance between the tick marks on the time ruler, where to the graphic window must be sized horizontally by using the mouse and pressing the SHIFT- key.

The graphic not only can be saved, but also printed out or copied to clipboard.



## ▶ Clip

### **In this dialog you can set the codecs that windows shall use for playing a clip**

Videograph and all relevant programs for playing media files use a multimedia interface of windows called "DirectShow". DirectShow components - sometime called "filters" - are registered in the system and are then available for all programs.

A whole chain of such filters is involved when processing audio and video data streams, whereby each manages a certain task. The media players then are obliged to manage the filter chain, having only indirectly anything to do with the processing of the data streams.

The DirectShow Filtermanager normally determines that the necessary filters are implemented. E.g., if a video file is to be played, the manager is assigned to determine the necessary filters and to develop the chain - the so called "filter graphs". Since this mechanism doesn't always lead to the desired result, programs can affect the process, requiring for example that certain filters are used in any case. Otherwise the manager takes the "best" filter found in the system for the task. The decision as to which filter is used is based on a value assigned to each filter, which is called "Merit".

However, since the Merit is specified by the manufacturer of the filter (who may tend to exaggerate, as one can imagine), this system does not always lead to an optimal result. Fortunately however, there are possibilities of changing the Merit.

There are several merit usage classes (these numbers are in hex):

MERIT\_PREFERRED = 0x800000

MERIT\_NORMAL = 0x600000

MERIT\_UNLIKELY = 0x400000

MERIT\_DO\_NOT\_USE = 0x200000

You can use any number above the class while still maintaining within the class, for example a merit of "400005" is still "Unlikely". By rearranging the merits of filters you can eventually resolve most conflicts. To change a filter's merit in videograph, follow the instructions given here.

#### **Problems with filters**

With most of the numerous media formats on the market, there are no playback problems, if there are, the cause is usually:

There is no suitable filter in the system for the media format of the clip to be played. In this case, either an error message is issued and there is no picture or sound or both, or the manager selects another similar filter as an alternative, which may not play the clip correctly. In such cases, you can usually find a suitable filter on the Internet and upgrade it. As already described in the chapter on "System requirements", the LAV filters are the optimal alternative for this purpose.

You should install the LAV filter first if a clip causes problems when playing in Videograph. As experience shows, the playback problems usually also disappear afterwards. The reason is that Videograph recognizes when the LAV filters are installed and can therefore set up the video configuration so that Windows always uses the LAV filters instead of Windows own codecs or any other codec when playing a video clip

Another possible cause is, that several filters are installed, which are declared as suitable for the media type of the current clip, whereby for some reason the filter that handles the media format of the clip best has a lower or only the same merit as the other filters, although these are less suitable. However, because of the poorer merit, Windows will probably not use the better filter.

With the merit dialog it is possible to change the merit of each filter, so you can downgrade the merit of undesirably used filters and upgrade that of the favored filter so that the merit of the favored filter is significantly higher than that of the competing filter.

A detailed description of the procedure with illustrations can be found in the Videograph online help. .

## ▶ Workspace

### Using multiple monitors


On many portable computers and many desktop computers (those with two video ports on one video card), you can expand your display to a second monitor and thus expand the size of your desktop. On such an expanded display you can stretch (by mouse) the main window of Videograph across two monitors, and so you can drag and move the diverse windows inside of the main window across the monitors as you please. You can e.g. place the video window of a clip on the first monitor and the other windows on the second, a good cause if several videos concurrently are to observe.

Please note that the left portion of the main window always appears on the monitor that serves as the primary display.

In order to be able to stretch the main window across the expanded desktop, the window must **not** be maximized. A maximized window has a fixed size, it will fill the area of the primary monitor only, you can't move a maximized window between screens or stretch it across several monitors. You must first reduce the size of the window either by clicking the Restore Down button on the window's title bar or by double-clicking the title bar.

Use Display in Control Panel to configure the settings for multiple monitors.

## ▀ Keyboard shortcuts in the timeline window

Key(n)	action
SPACEBAR	Play/Pause
CURSOR DOWN	scroll databars upward
CURSOR UP	scroll databars downward
click left mouse key while "M" key is pressed	Set a jump label at the current position of mouse pointer in the time ruler
click left mouse key while DELkey is pressed	Delete jump label at the place where the mouse pointer is
CURSOR RIGHT	wind forward clip by 1 second
CURSOR LEFT	rewind clip by 1 second
CURSOR RIGHT and SHIFT	wind forward clip to next tick mark
CURSOR LEFT and SHIFT	rewind clip to next tick mark
SHIFT- key with right clicking on the head of databars	Fade-in variable label
only if there is an <b>active</b> databar:	
TABULATOR	move up following databar
TABULATOR+SHIFT	move down active data bar
otherwise:	
TABULATOR-key	activates the coding - and transcript window
only if playback interval is <b>activated</b> :	
HOME (POS 1)	Play from start of interval
SHIFT and mouse clicking 	shift playback interval to start position 0:00:00
only if <b>jump markers(s)</b> set:	
NUMPAD*-	move clip to preceding jump label
NUMPAD*+	move clip to proximate jump label

\*NUMPAD = numeric keypad

## ▶ shortcuts in the coding window

Key	outcome
SPACEBAR	replay/pause
CURSOR UP	scroll coding bars (alternative: categories) upward
CURSOR DOWN	scroll coding bars (alternative: categories) downward
CURSOR RIGHT	wind forward clip by 1 second
CURSOR LEFT	rewind clip by 1 second
SHIFT + CURSOR RIGHT	wind forward clip to next tick mark
SHIFT + CURSOR LEFT	rewind clip to next tick mark
only if coding bar is <b>activated</b> :	
TABULATOR	move up following coding bar
TABULATOR+SHIFT	move down active coding bar
o otherwise:	
TABULATOR	activates timeline window
CTRL + M	set jump label at current time position
only if playback interval is <b>activated</b> :	
HOME (POS 1)	play from start of interval
CTRL + CURSOR RIGHT	move playback interval forward
CTRL + CURSOR LEFT	move playback interval backwards
only if databar is <b>activated</b> and playback interval is <b>deactivated</b> (see <a href="#">here</a> ):	
PAGE UP	wind forward to following coding interval
PAGE DOWN	Rewind to preceding coding interval
<b>CTRL</b> +PAGE UP or CTRL + CURSOR RIGHT	wind to end of current coding interval
<b>CTRL</b> +PAGE DOWN oder CTRL + CURSOR LEFT	wind to beginning of current coding interval
only when a <b>master-variable</b> is used:	
BILD OBEN	move playback area to following coding interval
BILD UNTEN	move playback area to preceding coding interval
only if <b>jump label(s)</b> set:	
NUMPAD*	move clip to preceding jump label
NUMPAD*	move clip to following jump label
only if there are <b>jump label(s)</b> and if playback area is <b>activated</b> :	
NUMPAD*	as above, but only if jump label lies within playback area
NUMPAD*	as above, but only if jump label lies within playback area
only if <b>write permission</b> and there are an <b>active coding bar</b> :	
KEYS 0-9 oder NUMPAD*0-9	numerical input of coding value
only if <b>write permission</b> and playback area is <b>deactivated</b> :	
'T'	insert out-point
CTRL+'T'	insert new in-point
debugging-function:	
ALT +'r'	rectify codingbar layout

NUMPAD = numeric keypad

► **Keyboard shortcuts in transcription window**

key(s)	alternatively	action
F5 (reserved but modifiable)		Inserts a carriage return at the caret position followed by the string "teacher:"
F6 (reserved but modifiable)		Inserts a carriage return at the caret position followed by the string "student:"
F12 (reserved but modifiable)		Inserts (at the caret position) a string containing the current video time
F2 - F12, beside F10		Insert a user-defined string at the caret position
<b>CTRL+ 'M'</b>		use to set or drag a <b>transcription marker</b>
CTRL+NUMPAD '+'	SHIFT+CTRL + '+'	move clip to proximate jump label
CTRL+NUMPAD '-'	CTRL +SHIFT + '-'	move clip to preceding jump label
only if <b>transcription marker</b> set		
CTRL+ENTF		erase <b>transcription marker</b>
NUMPAD '+'	CTRL + '+'	move forward (inside the playback area) <b>transcription marker</b> by 1 second
NUMPAD '-'	CTRL + '-'	move backward (inside the playback area) <b>transcription marker</b> by 1 second
NUMPAD ENTER	CTRL + 'H'	play from <b>transcription marker</b> (if existing)
CTRL+NUMPAD ENTER	SHIFT+ CTRL+ 'H'	play from start of interval
only if playback area is <b>activated</b> :		
CTRL+SPACE		play/pause
NUMPAD '0'	CTRL + 'P'	play/pause
NUMPAD ENTER	SHIFT+ CTRL+ 'H'	play from start of interval
NUMPAD '6'	CTRL + 'F'	wind forward clip by 1 second
CTRL+NUMPAD '6'	SHIFT+ CTRL + 'F'	move forward playback area
NUMPAD '4'	CTRL + 'B'	rewind clip by 1 second
CTRL+NUMPAD '4'	SHIFT+ CTRL + 'B'	shift back playback area
only <b>inside</b> of active playback area:		
NUMPAD ENTER	CTRL + 'H'	play from start of interval ( <b>or</b> play from transcription marker if existing)
NUMPAD '0'	CTRL + 'P'	play/pause
CTRL+NUMPAD '+'	SHIFT+ CTRL + '+'	go to proximate jump label inside the playback area
CTRL+NUMPAD '-'	CTRL +SHIFT + '-'	go to preceding jump label inside the playback area
only if <b>write permission</b> :		
CTRL+'T'		begin new turn
only if write <b>permission</b> and playback area <b>activated</b> :		
SHIFT+Einfg		Insert text from clipboard at the caret position
CTRL+Einfg		Copy selected text to the clipboard
SHIFT+Entf		Cut selected text and copy it to the clipboard

NUMPAD = numeric keypad

## ► Customizing the work space

### Keyboard shortcuts for switching

Keys	activated window
Alt+1	Timeline window for Video 1
Alt+2	Timeline window for video 2 (Alt+3= Video 3 etc.)
Alt+V+1	Video window for Video 1
Alt+V+2	Video window for Video 2 (Alt+V+3=Video 3 etc.)
Alt+C	Coding and transcription window
Tab key ( <b>within timeline window</b> )	Activating the coding and transcription window
Tab key ( <b>within coding window</b> )	Activating the timeline window