PAINT

Paint 10.0 Installation Guide

Release June 2025

Last Update: June, 2025

Table of Contents

Installation

Hardware

Chyron built hardware qualifies for running Paint 10. Here is the list of recommended hardware:

Mainboard

Supermicro X13SWA-TF

CPU

- Intel Xeon w5-3435X

Memory

- 64GB DDR5 4800Mhz

Graphics

- NVIDIA RTX A5000

SSD's

- 3x Micron 7450 MAX U.3 3,2TB 15mm Gen4 x4
- See Media drives

Installation on MS Windows

Paint 10 is certified to run on:

- Microsoft Windows 10 Professional x64 version 22H2 (OS build 19045)[1]
- Microsoft Windows 11 Professional x64 version 23H2 (OS build 22631)[1]

Anti Virus

On the Paint 10 machine, it is recommended to not use Windows Defender or any anti-virus program that performs a real-time scan of the system since this influences the system performance. If necessary run manual scanning of the system when the system is not on-air.

Windows services

Windows Indexing Services and Windows Update Services should be disabled.

UAC settings

Disk recorder video file allocation needs elevated user privileges. The allocation is performed first time the application starts or after the video file size is changed in application config.
To meet that requirement, either:

- start the application as an administrator (right-click application icon, select Run as administrator)
- set the UAC level to 'Never notify' at the User Account Control Settings dialog (Control Panel → User Accounts → Change User Account Control Settings)

Windows firewall

It is recommended that the Windows firewall is enabled with exception or rule for dedicated UDP port using for remote control of Paint.

Windows 11 notice

Windows 11 upgrades with an official image for Paint are handled by sales. Please contact your sales representative to move to a verified Windows 11 image. This image is only tested with Matrox i/o boards.

[1] This page describes how to detect your Windows version

NVIDIA Display Drivers

Systems equipped with NVIDIA GPU require following display drivers installed.

Card Type	Drivers
GeForce family graphics cards support according to nvidia driver release notes	537.13
Quadro family graphics cards support according to nvidia driver release notes	537.13

NVIDIA website**:** http://www.nvidia.com

Setting

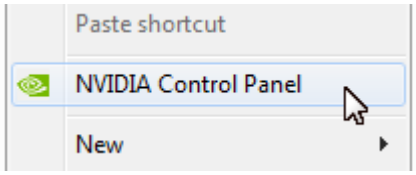
NVIDIA graphics drivers must be properly configured for running Paint.
Open Control Panel-Display-Settings and set Color quality to 32 bits. Driver setting is done automatically using NVIDIA applications profile for Paint 10 during application installation.

Important

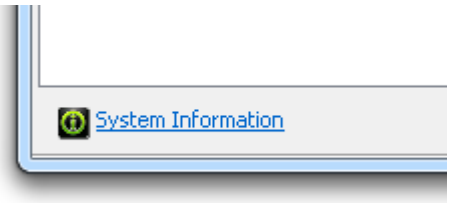
You must reinstall Paint after upgrading or reinstalling the NVIDIA driver!

How to know what is the current driver version

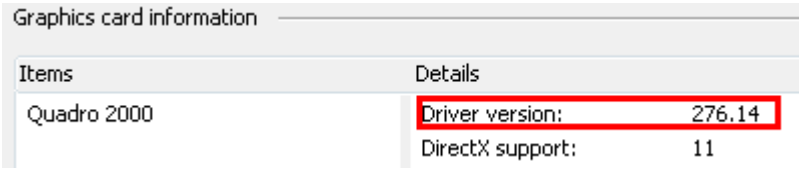
- Do right mouse click on an empty desktop. A popup menu appears. Select "NVIDIA Control Panel".



- Click "System Information" in the left-bottom corner:



- Read the GPU name in the left column and driver version in the right column.



UHD Video Modes

UHD (4K) video modes require a more powerful GPU card than normal HD. Minimal requirement is NVIDIA Quadro P4000.

Intel display drivers

Systems equipped with INTEL GPU (UHD graphics 620 and newer) require following Intel drivers installed: `31.0.101.3790/31.0.101.2114`.

Intel website: http://www.intel.com/

Matrox Video Card

Supported video cards

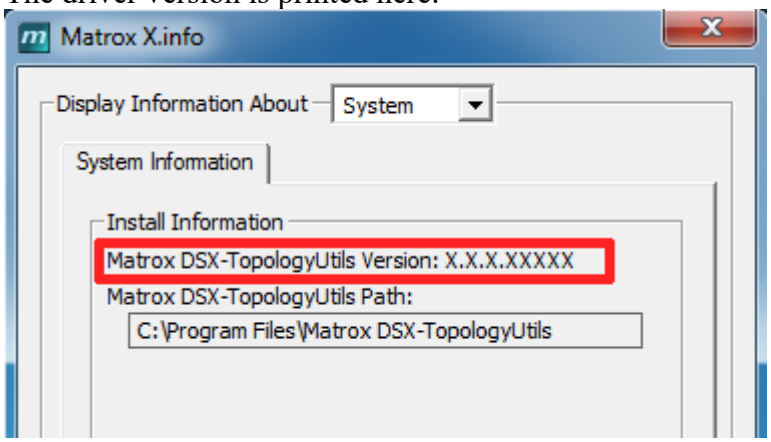
- DSX LE4 LP /4
- DSK LE4 LP /8
- DSK LE4 FH /4
- DSX LE4 FH /8
- DSK LE4 FH /X2
- DSX LE5 12G

Driver

Driver **10.5.101** has to be installed. This driver is available on https://da.chyronhego.com/ in Paint installer section or **Third Party SW**. Other drivers are not supported!

If you are not sure which video card or driver version is installed then start Matrox X.info `"C:\Program Files\Matrox DSX-TopologyUtils\System64\mveXinfo.exe"`. It is started automatically. See this icon in taskbar:

The driver version is printed here:



Video IO configuration selection

The video board supports multiple input/output configurations. The current setting is visible in the X.info application. See screenshot above. Look for "SDI in: Out: ".

Supported configurations

LE4 LP /4	LE4 LP /8	LE4 FH /4	LE4 FH /8	LE4 FH /X2

2in2out 3in1out	4in4out	2in2out 3in1out	4in4out	8in4out 4in8out

The configuration is programmed in the board and can be changed by supplied command line tool:
`"C:\Program Files\Matrox DSX-TopologyUtils\drivers\mvConnectorConfig.exe"`

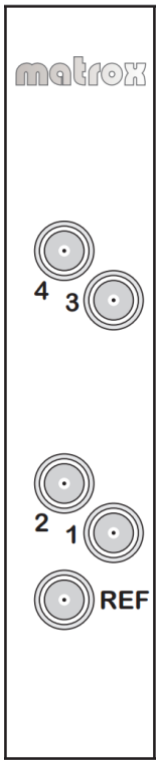
For example, if you want to change the configuration to 3 inputs and 1 output on LE4 LP /4 board you do this:

- Open command prompt (start cmd.exe)
- `cd "C:\Program Files\Matrox DSX-TopologyUtils\drivers"`
- `mvConnectorConfig.exe" -3in1out`
- If you have multiple cards you have to set them individually by adding serial number after the configuration type and separated by character =. E.g.
  `mvConnectorConfig.exe -3in1out -sn=A530986`
- The configuration process takes a while. When it finishes you are asked for a PC reboot to apply the changes.

Supported video formats

- SD 576i 50, SD 486i 60M
- HD 1080i 50 or 60M
- HD 720p 50, or 60M
- 3G 1080p 50 or 60M
- 4K 3840x2160 50, 60, 60M

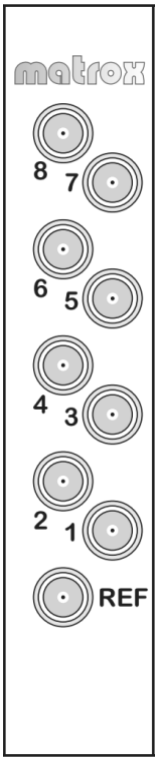
LE4 LP/4 connectors



Connector	Configuration 2in2out	Configuration 3in1out
4	SDI Out 1 or Key Out 0	SDI Out
3	SDI In 1	SDI In 2
2	SDI Out 0	SDI In 1
1	SDI In 0	SDI In 0
REF	Reference input	Reference input

LE4 LP/8 connectors

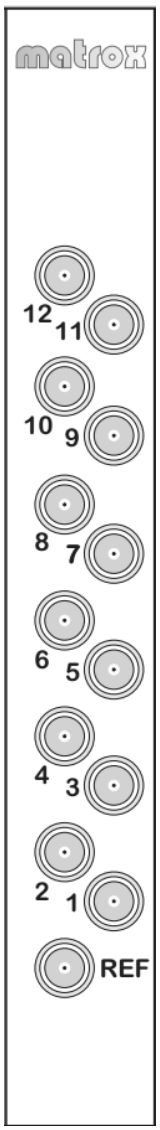
One 4K video uses 4 connectors.



Connector	Configuration 4in4out
8	SDI Out 3
7	SDI In 3
6	SDI Out 2 or Key Out 0
5	SDI In 2
4	SDI Out 1
3	SDI In 1
2	SDI Out 0
1	SDI In 0
REF	Reference input

LE4 FH connectors

One 4K video uses 4 connectors. The first 4K input connectors must be either In 0 or In 4.



Connector	Configuration 4in4out	Configuration 8in4out	Configuration 4in8out Fill/Key in 4K
12		SDI In 7	SDI Key Out 3
11		SDI In 6	SDI Key Out 2
10		SDI In 5	SDI Out 3
9		SDI In 4	SDI Out 2
8	SDI Out 3	SDI Out 3	SDI Key Out 1
7	SDI In 3	SDI In 3	SDI In 3

6	SDI Out 2 or Key Out 0	SDI Out 2 or Key Out 0	SDI Key Out 0
5	SDI In 2	SDI In 2	SDI In 2
4	SDI Out 1	SDI Out 1	SDI Out 1
3	SDI In 1	SDI In 1	SDI In 1
2	SDI Out 0	SDI Out 0	SDI Out 0
1	SDI In 0	SDI In 0	SDI In 0
REF	Reference input	Reference input	Reference input

Matrox M264 codec card

Matrox M264/S2 card is required for real time H.264 codec in 4K (UHD). No other codec setting is sup

Media drives

Paint requires dedicated media drives for storing video. It is not allowed to use a drive that is accessed by other applications during Paint operation.

You can choose from several approved setups. The total media drive performance requirement depends on the number of cameras and video format.

Supported configurations are

- 1-6 HD cameras
- 1-3 4K cameras

Video file size

The video file stores video and audio for clips and record train for all input cameras. Following table shows expected file size in GB (1024^3) to store 30 minutes of video.

Video Format	1 camera	2 cameras	3 cameras
HD 1080i 50Hz	32	64	96
HD 1080i 59Hz	37	74	111
HD 1080p 50Hz	64	128	192
HD 1080p 59Hz	74	148	222
HD 720p 50Hz	35	70	105
HD 720p 59Hz	43	86	129
4K 2160p	120	240	360

The table above assumes JPEG compression for HD/3G video, quality set to 95%. For 4K hardware codec M264 is required with "M264 XAVC 4K Intra 480 CBG 10Bit" profile.

The longer video DAT file the longer time it takes to start the Paint. It is not recommended to store more than 5 hours of video.

SSD drives

Always leave at least 20% of free space on a SSD drive! Do not fill up the space with either the main video dat file or with any other files. The 20% of the total space has to be just empty. SSD requires this empty space to be able to achieve high sustained write speeds.

Microsoft Windows Notice!

Disk recorder video file is not supported to be located on C: drive.

Jog/Shuttle controller

Please install appropriate driver downloadable at the ChyronHego download area, e.g. `cdi_shuttle_win_2.81.exe`.
If you're using Windows 10 x64 operating system, please restart your computer after the driver installation.

Sentinel key USB driver

Please install USB driver for Sentinel dongle with Paint 10 license.
The current version is at Hego Download Area as a third party software in file `Sentinel_System_Driver_Installer_7.5.8.exe`

Java

Since Paint 8.5, Java runtime is distributed along the application and does not have to be installed on system prior using the application.

Windows Settings

Anti Virus

On the Paint 10 machine, do not run any antivirus programs that perform a real-time scan of the system since this influences the system performance. If necessary run manual scanning of the system when the system is not on-air.

Windows services

Do not enable Windows Indexing Services.

UAC settings

Disk recorder video file allocation needs elevated user privileges. The allocation is performed first time the application starts or after the video file size is changed in application config.
To meet that requirement, either:

- start the application as an administrator (right-click application icon, select Run as administrator)
- set the UAC level to 'Never notify' at the User Account Control Settings dialog (Control Panel → User Accounts → Change User Account Control Settings)

Windows firewall

It is recommended that the Windows firewall is enabled with exception or rule for dedicated UDP port using for remote control of Paint.

Folder Structure

Paint uses the folders as follows:

- Application folders:
  `C:\Program Files\ChyronHego\Paint10\Paint.app`
  `C:\Program Files\ChyronHego\Paint10\ConfigEditor.app`
- Data folder (containing clip data file, tools, skins, etc.):
  `Documents\Paint9`
- Log folder:
  `C:\log\ChyronHego\`

Software Installation

To install Paint 10 please run PaintSetup<*version*>.exe

This installer will guide you through the installation process.

Software updates

Paint 10 can be upgraded when the new version is released. All your configurations are preserved.

Each release has it's version. Major number . Minor number . Revision number.
For example 7.0 or 7.0.2. Updates done in releases with changed revision version number are bug fixes or very small improvements. Change in minor or major version number indicates bigger changes.

You should always read Release Notes before upgrading the application. If your application is running fine, do not upgrade to a new major version.

All updates are available at ChyronHego Download Area, https://da.chyronhego.com/
Please ask ChyronHego support to obtain a login email and password to access this site. You can register your email address to receive notification emails about new releases.

Silent Installer mode

In Silent mode, the installer does not show a step-by-step dialog. Instead, it runs without any user interaction, using default installation options. This way of installing Paint is suitable for automated environments.

MS Windows
To start installation in silent mode, start the installation process from command prompt, using following parameters:

VERYSILENT	Prevents using any installer GUI.
SUPPRESSMSGBOXES	Prevents showing any question messages, assuming reasonable default responses from the user.
LOG="*logfile*"	Logs installation progress into specified file.
NORESTART	Prevents the Operating System from being restarted after installation finishes.

For detailed info on possible installer parameters, see InnoSetup Command Line Parameters.

Example:

```
PaintSetup9.8.exe /VERYSILENT /SUPPRESSMSGBOXES /LOG="PaintInstLog.txt"
```

License

The license is programmed into the Sentinel USB dongle. The dongle is required to run Paint 10 and it has to be present during the application operation. Removing the key might shut down Paint 10.

The License dongle must be inserted in the USB port prior to Paint installation.

There are 2 types of license keys supported:

SHK

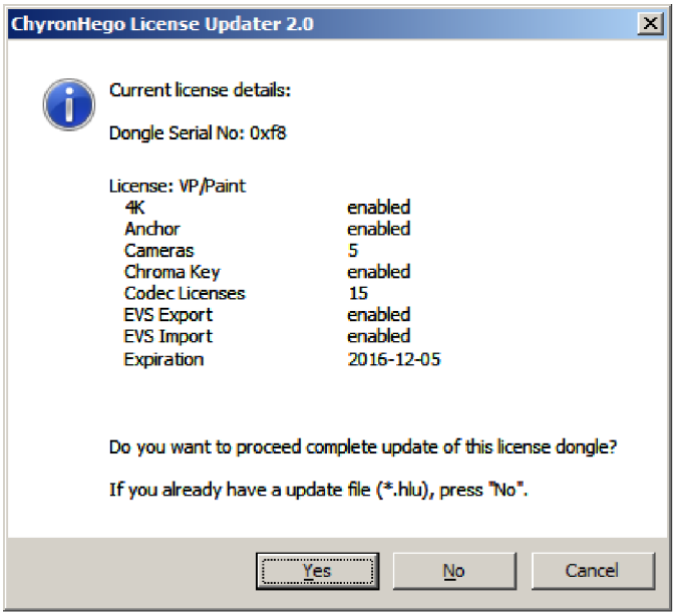
HL

License update

MS Windows

Run Start → Paint 10 → ChyronHego License Updater.

The updater shows all licenses present in the license key. Press Yes to run the update. Connection to the Internet is required.



Version licensing

Major versions of Paint are licensed separately requiring purchase of new license followed by an key update before use.

Performance monitoring support

To easily identify possible performance bottlenecks, Paint 10 publishes numerous runtime parameters. By default, the app is using browser-based monitoring.

Displaying Paint performance counters

To immediately view Paint performance counters, locate the URL defined in *Config Editor's General page → Watch Publisher URL* and input it to a web browser.

Windows Performance Monitor

Previous form of Windows Performance Monitor counters can be enabled by renaming the library *WatchPublisherWinCounters.dll* to *WatchPublisher.dll* and then putting it in the root installation directory instead of the existing one.
Performance Monitor tool can be then found in the Start *menu → Control Panel → Administrative Tools*. Alternatively, you can enter the phrase 'perfmon' into the Start menu search box and hit Enter.

After opening the WPS window select *Performance Monitor* in the *Performance → Monitoring Tools*. A graph view shows up. Press the right mouse button over the graph area and select Add Counters... in the popup menu. In the *Available counters* list, locate and add ChyronHego categories. Paint application must be already running when adding counters to graph.

Recording Paint performance counter logs for later analysis

To record performance counter log, locate *Data Collector Sets → User Defined* node in the Performance Monitor tree. Click the right mouse button over the Paint set and choose Start in the popup menu. Choosing the Stop menu item finishes data collection.
Performance log files are stored in the *C:\log\ChyronHego* directory.

Troubleshooting

However, we make an effort to eliminate any errors in the application, sometimes problems can occur. Installation of Paint 10 includes a diagnostic tool, which gathers system information and log files, configuration files and other auxiliary setting files. It helps us to determine a reason. Diagnostic Tool service offered automatically after the crash of the application has been detected. Alternatively, if the user experiences any problem, it is possible to launch the Diagnostic Tool from the Start menu - Paint 10 - ChyronHego Diagnostic Tool and follow the instructions. The output compressed zip file can be directly uploaded to our support center or sent via email.

Since Paint version 8.1.4 the TeamViewer QuickSupport is also available, enabling real time remote assistance from one of our specialists for help with serious or time-sensitive issues.

Adjustable error log verbosity

You can adjust the verbosity of logging for different modules of the application. It can be done by switching type of logging inside the config editors Logging tab.

Module attributes

Attribute	**Description**
name	Name of the application's module.
value	Type of log filtering. Can be one of these: ● DEBUG - the most detailed, all types of logs are logged ● INFO - only info, warning and error logs are logged ● WARNING - only warning and error logs are logged ● ERROR - only error logs are logged

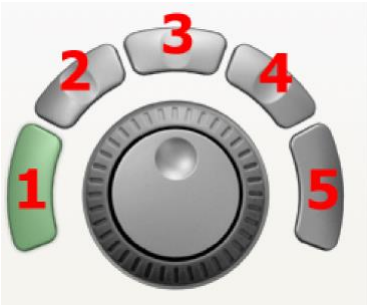
Contour ShuttleXpress / ShuttlePro2 / ShuttlePro controllers

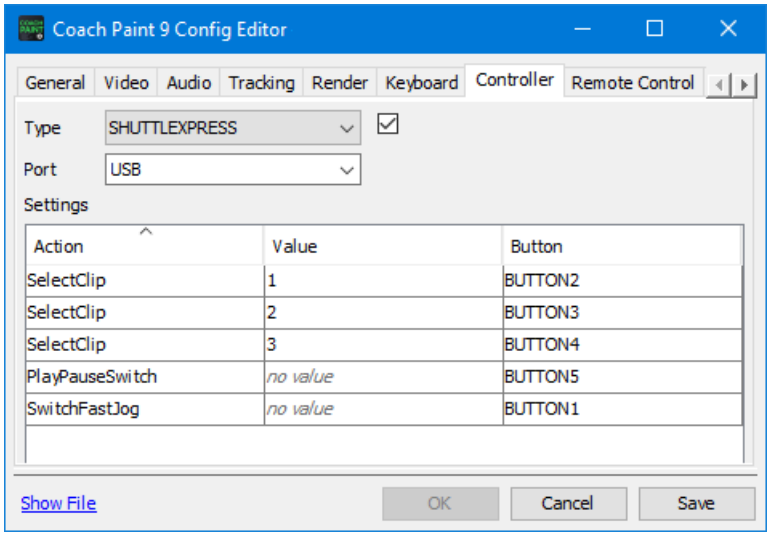
Many Paint 10 functions can be controlled by these controllers. The controller buttons can be mapped to trigger actions in Paint.

Windows

To assign (or change) the function of a button on Windows open the **Config Editor** in the Controller tab.

ShuttleXpress button assignment





Default ShuttleXpress settings

Button	Action Description	Action
1	Switch FastJog on/off	SwitchFastJog
2	Clip 1 in currently selected bank	SelectClip 1
3	Clip 2 in currently selected bank	SelectClip 2
4	Clip 3 in currently selected bank	SelectClip 3
5	Switch play/pause (according to actual state)	PlayPauseSwitch
Shuttle ring	Shuttle forward / backward	ShuttleForward 0.25 – 8 ShuttleBackward 0.25 – 8

Remote Control using Action based protocol

The application can be remotely controlled via XML-based protocol with a set of supported actions. Connection should be established using TCP or UDP protocol. TCP and UDP have the same capability.

Action based protocol allows other software to control Paint application.

Message format

Format of the data sent over the TCP/UDP channel is plain text XML.
Example:

```
<Pause/>
```

Example of action with parameter:

```
<Play value="1"/>
```

UDP connection requires one message per datagram. TCP connection expects the messages terminated by null character.

Example of configuration:

```
<RemoteControl>
 <Enabled value="1"/>
 <Port value="UDP5000"/> <!--UDP at port 5000, use e.g. TCP5000 for
TCP protocol -->
 <Protocol value="ACTION"/>
</RemoteControl>
```

Available actions are described in the file doc/Actions.html (or Start menu → Paint 10 → Documentation → Actions)

Remote Control using REST protocol

Clients can communicate with Paint using a REST protocol. The protocol allows to trigger user actions or query Paint runtime values.

Configuration

Allows to specify port used for REST commands.

```
<RemoteControl>
  <Enabled value="1"/>
  <Port value="8000"/>
  <Protocol value="REST"/>
</RemoteControl>
```

REST action invocation

To trigger a user action, send a http POST request to an url in form:

`http://<paint.host.name>:<REST.port>/action`

Request body must contain xml with action data formatted the same way as Action-based protocol.

Example:

`POST http://localhost:8000/action` req data `<Clear/>`

REST value queries

To query runtime value, send a http GET request to an url in form:

`http://<paint.host.name>:<REST.port>/query/<valueName>[?param1[&param2]]`

The query can contain additional parameters in form: `<param.name>=<param.value>`

Example:

```
GET
http://localhost:8000/query/Clips?from=123&to=345
```

Paint sends a reply containing a xml data formatted as:

`<QueryResult query="<valueName>">valueData</QueryResult>`

Value data can be either a string representation of primitive value or a xml structure in case of non-trivial values.

In case the query can not be resolved, a 404 error is returned.

Values supported by REST query

Value Name	Value Description	Reply Description
PreviewCamera	Camera ID currently selected in preview.	32bit integer
PreviewTimecode	Timecode of frame currently shown in preview.	64bit signed integer
VideoFormat	Current video format: • id - format name (e.g. HD1080P_50) • aspect - “4:3” or “16:9”	`<QueryResult query="VideoFormat"> <id value="HD1080P_50"/> <aspect value="16:9"/>`

	• width - image width in pixels • height - image height in pixels • interlace - true / false • fps - number of fields/frames per second as a floating point number	<width value="1920"/> <height value="1080"/> <interlace value="false"/> <fps value="50"/> </QueryResult>
Clips	Retrieve a currently stored clips: • name - name of clip • from - first timecode of clip • to - last timecode of clip • cameraId - id of camera	<QueryResult query="Clips"> <Clips> <Clip name="clip0_0|CLIP10" from="3536828" to="3537885" cameraId="120"/> <Clip .../> ... </Clips></QueryResult>

Remote Control using Web browser

The application can be remotely controlled via a web browser.

Remote control via web browser requires Remote Control license.

To enable preview remote control, launch Config Editor, locate Remote Control tab and enable Web Control checkbox. Output Streaming feature gets enabled automatically.
Web page is then accessible via browser at:
"http://" +computerLocalAddress +":" +httpPort (as defined in config, default: 10127).

Example:
http://192.168.1.81:10127

It is possible to easily copy control URL to clipboard by clicking the Config Editor URL label.

Web elements description

Element Name	Description	Value
Enabled	Enable/disable web module.	0 - disabled, 1 - enabled
HttpRoot	Path to root directory of web page, set by	A string, set by installer

	installer	
HttpPort	Defines port web page is provided at	A number defining port
WebSocketPort	Port used for bi-directional communication via websocket with web page	A number defining port

Playback Settings

It is possible to configure some global playback settings. Values are defined in config.xml, see example below:

```
<ClipPlayback>
      <ShuttleSpeed value=”2”/>
      <FastJogCoefficient value=”50”/>
      <FastShuttleCoefficient value=”4”/>
</ClipPlayback>
```

Value	Description	Default
ShuttleSpeed	Playback speed reached when shuttle control is turned to its maximum angle.	1
FastJogCoefficient	Multiplication of jog step with fast jog enabled	50
FastShuttleCoefficient	Multiplication of shuttle speed with fast jog enabled	4

Font Engine Selection

Paint supports two font rendering engines:

Legacy - default, can be used in most situations
DirectWrite - must be used in order to get right-to-left texts rendered correctly

Selection of font engine can be done in Paint Config Editor at the Render tab.

EVS Integration

Paint 10 can import/export clips from/to EVS XT. The import process is usually faster than real time. Clips are transferred over a Gigabit Ethernet network.

EVS Setup

If neither PC nor EVS XT is connected to the existing Gigabit network then assign an IP addresses of your choice. You should use an IP address in one of the private ranges. Example:

	EVS	**Paint**
IP Address	192.168.044.001	192.168.44.2
Subnet Mask	255.255.255.000	255.255.255.0
Default Gateway	192.168.044.002	no need to set

It is recommended to connect the Paint PC and the EVS XT by direct cable. If you cannot use direct link then use one of the network switches approved by EVS. See EVS's manual
XT Technical Reference – Hardware for more details.

Following EVS XT servers are supported:

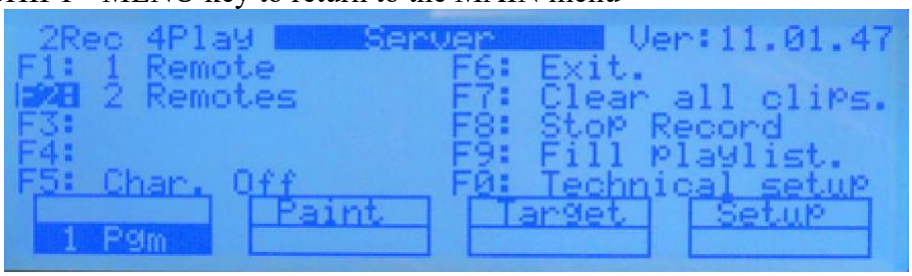
- XT[2]
- XT3
- XTnano

Important
It is required to use EVS software Multicam version between **10.03** and **15.3** (supporting FoxX)!

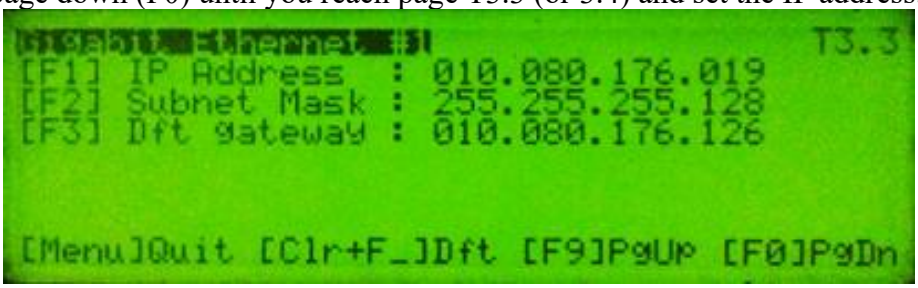
You can use any of the two Gigabit ports available on EVS XT.

Set IP address of the Gigabit Ethernet network on XT server.

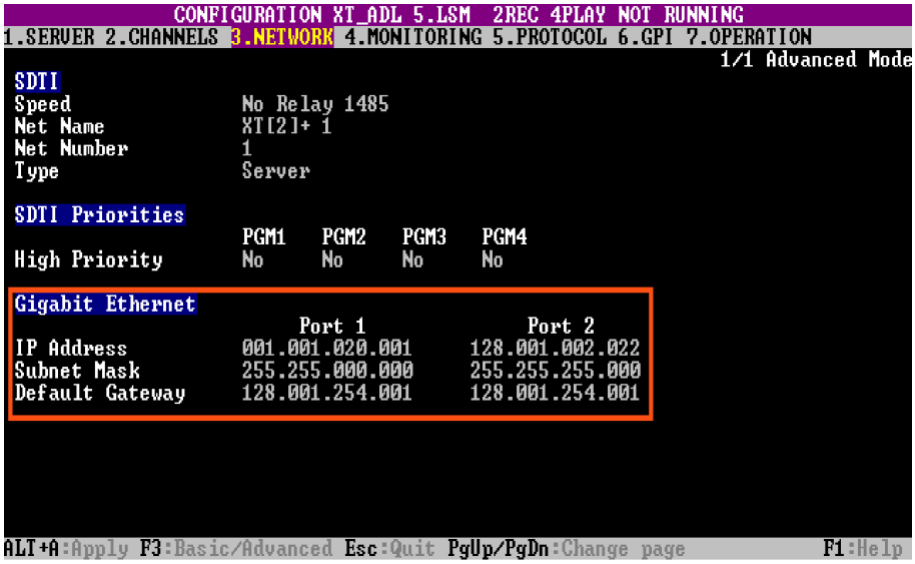
- Press SHIFT+ MENU key to return to the MAIN menu



- Press F10 to enter Technical menu (Shown an F0)
- Press page down (F0) until you reach page T3.3 (or 3.4) and set the IP address.



You can also do this via the VGA (Shift + F2 on the keyboard).

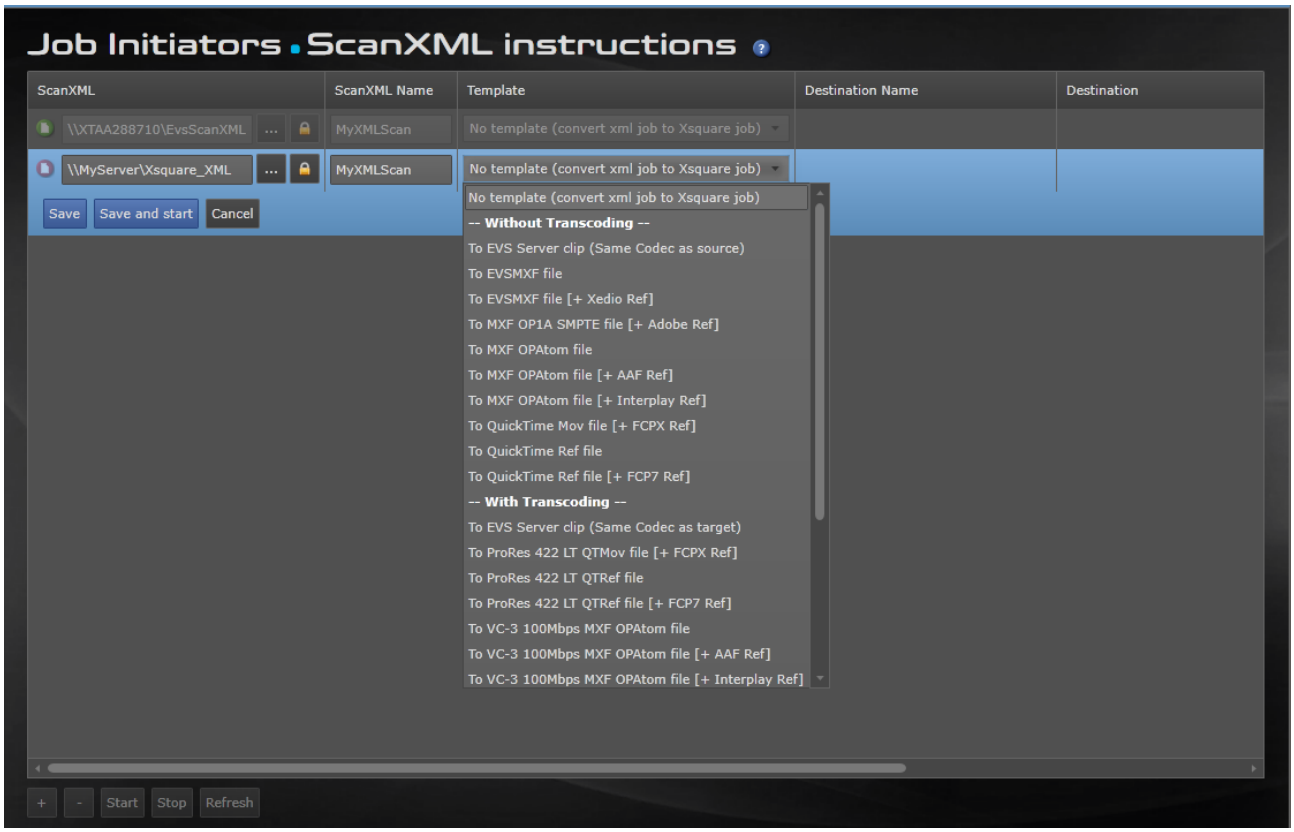


Important
For direct use of EVS software Multicam it needs to be between versions **10.03** and **15.3** (supporting FoxX)!

EVS XSquare

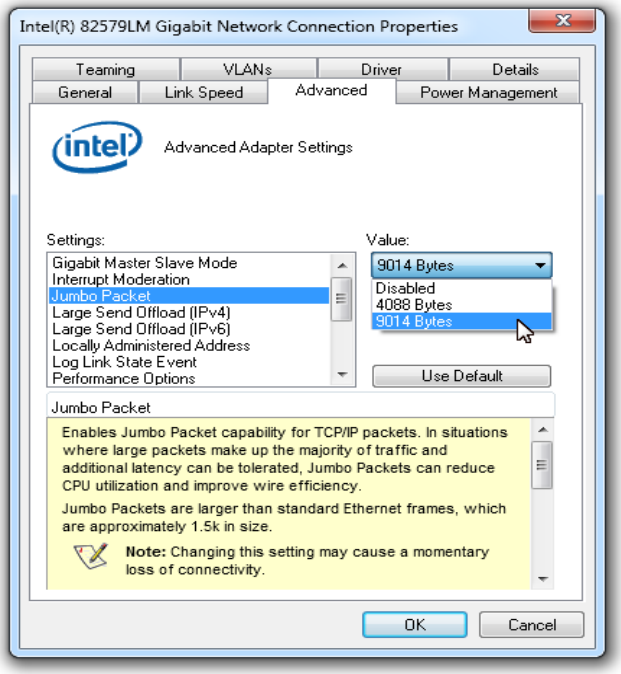
From EVS version **15.3** (not supporting FoxX) a XSquare server is needed to mediate communication. It needs to be connected to the EVS and have a running ScanXML Job Initiator configured as such:

- ScanXML - Path to network shared directory visible from both the XSquare server and the Paint client.
- Template - “No template (convert xml job to XSquare job)”



PC Setup

The network card and the driver are required to support jumbo frames. You need to enable the jumbo frames of at least 9000 bytes in the driver setting. By default the jumbo frames are disabled.



Go to the Control panel, Device Manager, select your network card and select Properties in pop-up menu. The Jumbo Frame setting is usually in Advanced tab.

Paint Configuration

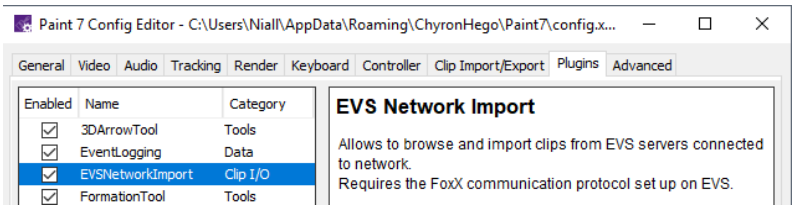
Paint needs to be configured to communicate with EVS XT server. Open Paint Config Editor, select category Clip Import/Export.

EVS Network support

Enabling EVS Network Access allows user to browse and import clips stored on EVS servers connected to the network. To use network access, EVS servers must fulfill following requirements:

- license code 123 installed
- network connection on PC LAN connector (browsing via LinX protocol)
- network connection on GigaBit Ethernet 1 connector (clip transfer)

EVS Network Access can be enabled by checking the Access EVSNetworkImport option in the Plugins tab.



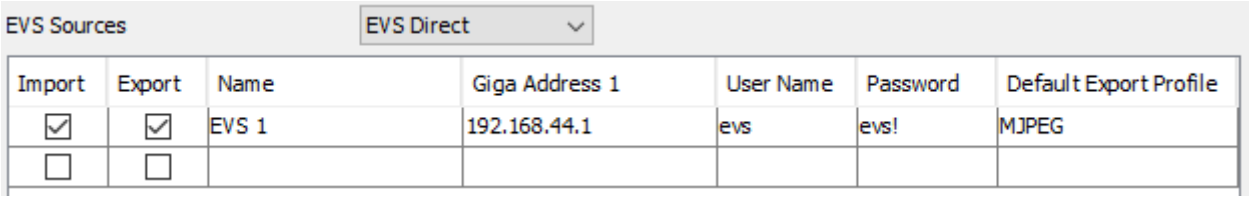
With EVS Network Access enabled, Paint import UI offers additional 'EVS Network' import source that provides additional Browse button when selected.

Manually configured EVS Sources

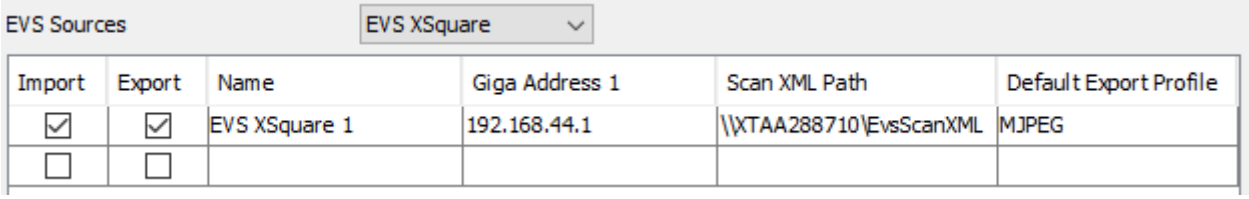
When EVS Network access can not be used for some reason or when there is a requirement for exporting clips to EVS, EVS servers can be configured as individual export/import sources.

You can define a list of your EVS LSM machines. The EVS machine name will appear in Paint UI during import. You will be able to choose from the list of configured machines. Choose the type of required source based on the EVS version (see Direct vs. XSquare) and create it. Each line has a setting for Import and Export. Only enabled machines show up in corresponding import or export dialog.
Giga Address 1 value needs to be set to the IP address of the 1st gigabit port of the EVS server. With XSquare you can find it in its configuration in section Monitoring / EVS Servers.

EVS Direct: Default user name is "`evs`" and default password is "`evs!`".



EVS XSquare: Default Scan XML Path is empty but needs to be configured to be the same as in the XSquare Job Initiator.

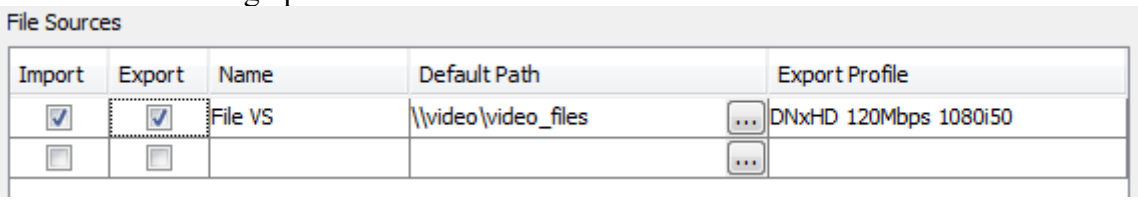


Advanced: You can set an environment variable EVS_SHORT_IN to 0 to import clips from Protect-In instead of Short-In. This setting is required when using standard PC ftp server as a LSM emulation.

EVS Import requires a license.
EVS Export requires a license.

File Clip Import/Export settings

Several file sources can be predefined in the config editor. The source names will be shown in the Import/Export dialog in the Source selector. Default path for each source can be set. This path is then used when the dialog opens.



Most commonly used video formats are supported. Mind, the video format of the clip and current video format currently set must match. Also check compatibility of export profile and currently selected video format.

Automatic Import From Watch Folder

You can set a folder to be watched for changes. When a clip file is copied into the folder by external application, it is automatically imported into Paint. The clip file has to have the same video format as actually set in the Config.

- When paint is launched, the Watch folders are scanned and all found clips are imported.
- When a new clip is copied into the Watch folder, it is imported.
- Once the clip is successfully imported, it is **DELETED** from the Watch folder.
- For network drives use UNC path (in form \\<PC_NAME>\<PATH>).
- You can specify a Clip button for the imported clip.
- VDR skins:
  - Clip named <bank_index>_<clip_index><.suffix> will be assigned to the button at <clip_index> within <bank_index> bank (e.g. Clip named `2_10.mov` will take place in the 2nd bank, 10rd position. Existing clip at that position is deleted. All indices are 1-based.)
  - In other cases the clip is assigned to the first vacant Clip button.
- Default skins:
  - Clip named <index><.suffix> will take place at the position given by the index in the clip bar. If there's a clip already at this position, it will be deleted.
  - In other cases the clip is imported but not added into the Clip bar. It is accessible in the Tool administration dialog and can be added manually.

Import/Export external metadata

Some systems may save clip metadata to external file. Paint is able to import such metadata together with the clip and export the file again without any knowledge of the metadata file. When a file with the same name (and specified extension) is found during import in the same directory as imported clip, it is imported.

Two types of metadata files are recognized by their extension: .xml or .xchange. Use Metadata File selector at Config Editor's Clip Import/Export page to choose metadata file type actually used.

Newtek Network Device Interface (NDI)

This part covers NDI usage and configuration.

General

NDI source naming standard is: `'MACHINE_NAME (SOURCE_NAME)'`

Eg.: say your computer is called Personal PC and you stream through VLC Media Player. Source will then be named '`Personal PC (VLC)`'.

Config editor setup for input

If you try connecting to a source, you only need to specify either source name (assuming there's only one source with this name in your network) or machine name (assuming there's only one NDI source streaming on your machine). If you want to be sure, specify both machine and source name.

Two input boxes (`NDI Source Input`, `NDI Machine Input`) get enabled when selecting NDI Video I/O HW. As mentioned above, at least one of them has to be set in order to work correctly.

If you're unsure about your source/machine name, write any name into source/machine input box, run the app and check Paint's log. Paint will try to connect to your source/machine but will probably fail (since the source is non-existent). It will however look for other available sources in your network and print their names into log. You can then specify the names correctly.

NewTek NDI

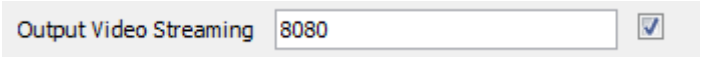
NDI driver is not bundled in Paint installations. For using NDI, please install NDI Runtime (driver) here (https://da.chyron.com/da/download-file.php?fillD=19947).

Config editor setup for output

Paint can also stream through NDI protocol. Specify source name in `NDI Source Output` text box.

Output video streaming

It's possible to stream video output as H264 MP4 stream. The only thing you need to do is enable output streaming in config editor and setup the right port (10126 by default). This option can be found in the video tab.



Video streaming feature is using either NVIDIA or Intel streaming technology. In case of Intel graphics hardware on Windows platform, driver 31.0.101.3790/31.0.101.2114 is required. For NVIDIA hardware, use driver according to NVIDIA graphics driver recommendation.

OSD Output

Since version 8.1, Paint allows to insert OSD information into the secondary SDI video output. Image is overlaid with textual data, containing:

- Time of Day of clip creation
- Total Clips Time/Clip Time remaining
- Clip Name
- Clip play rate

To enable the OSD output, locate Output selector at Video tab in the Config Editor. Choose *Main & OSD* value.

Audio Bars

It is possible to enable preview visualization of audio channels data in the form of audio bars rendered along the side of the video.
To enable Audio Bars in preview, go to the General tab in the Config Editor and enable the Audio Bars checkbox.

Tool Reference

This part covers tool usage and configuration.

Common Tool Configuration

Tool configuration is stored at by default in `Documents/Paint/Tools` folder. Each of the sub-directories contains a `tool.xml` file that configures the tool and it's resources. Beware the changes in `tool.xml` are applied only when the tool is drag/dropped to the tool palette. The Paint profile file (`.top`) in `Documents/Paint/Profiles` contains `tool.xml` content of all tools installed in the palette.

All `tool.xml` files support the following common tool parameters:
(parameters in italics are advanced)

Tools placed to the tool palette allow changing their configuration by clicking the right mouse button and selecting *Properties...* item in the popup menu. Changes in configuration are applied immediately when the property dialog is closed.

`Id`	Unique identifier of the tool
`In/OutEffectName`	Name of the In/Out effect.
`ChromaKeyEnabled`	Setting value 'true'/'false' ('1'/'0') enables/disables keying of the tool
`KeyframeAnimEnabled`	Setting value 'true'/'false' ('1'/'0') enables/disables tool keyframe animation.
`DefaultDuration`	Default length of the production. Infinite when disabled. Used in case that length of the production is not defined by keyframes.
`Thumbnail`	Thumbnail image displayed on the tool button in GUI.
`ThumbnailSelected`	Thumbnail image displayed on the selected tool button in GUI. If not defined, `Thumbnail` property is used for selected tool as well.
`ThumbnailColoringEnabled`	Setting value 'true'/'false' ('1'/'0') enables/disables application of Color value to the thumbnail.
`PlaceType`	*Tracking of the tool graphics. Values available are:* `aki.productplacement.place.screen.PlaceModelScreen` *(default)* `aki.productplacement.place.screen.PlaceModelScreenMotion`
`ClearOnPlay`	*Setting value 'false' ('0') disables application of global Clear on Play for this particular tool.*

Value types

- Duration is expressed in fields. For example 50 fields per second.
- Color is a vector of Red, Green, Blue and Alpha. Values are in range from 0.0 to 1.0.
- Opacity is a number from 0.0 (fully transparent) to 1.0 (fully opaque).

Tool Texture Settings

Most tools are rendering graphical elements covered with texture. There are two possible types of texture source that can be used with Paint:

static	• image file • most of raster image formats are supported (.png, .jpg)
dynamic	• proprietary .gtc file format • in combination with additional settings allows to produce advanced graphical effects

For any texture, a standard set of properties is available.

`File`	Name of the texture resource file (image or .gtc clip)
`PauseFrame`	Index of the texture frame to pause playback at. Applied when `PlayIn_Pause_PlayOut` PlayMode is set. Ignored for image-based textures.
`LoopStart`	Index of the .gtc clip frame to wrap to on loop. Used when `Loop, PlayIn_Loop` play mode is set. Ignored for image-based textures.
`PlayMode`	Decides how the .gtc clip is played on production. One of `Loop, PlayIn_Loop, PlayIn_Pause_PlayOut` values. See GTC Texture Play Modes for details.

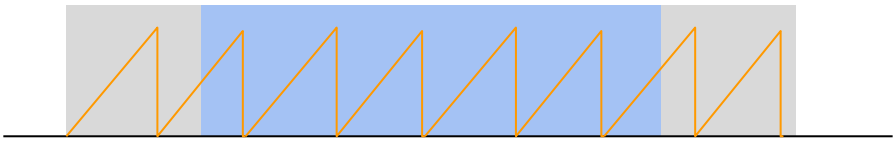
GTC Texture Play Modes

The `PlayMode` property gives user precise control over the presentation of the .gtc-based texture while the clip is being played. A detailed description of individual modes follows.

Diagrams are showing timeline overview, with following colors:

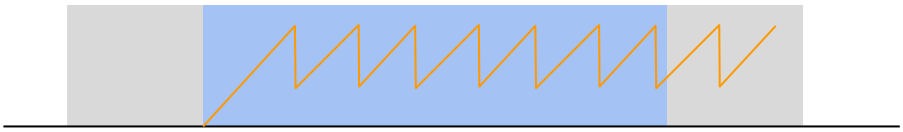
	Paint clip range (between in- and out- point)
	Drawing production range
	.gtc clip frame being displayed

Loop



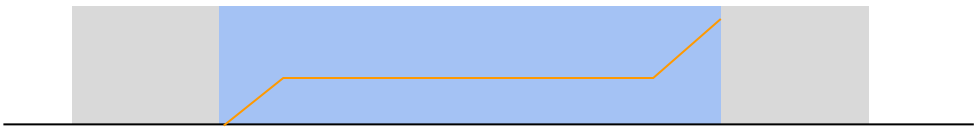
Default play mode for most tools. Start of the texture playback is not synchronized with the start of the graphics production. As a result, the texture shows an arbitrary clip frame at the beginning of the production.

PlayIn_Loop



Allows to synchronize the start of the texture playback with the beginning of the production. Useful in combination with the `LoopStart` property to use .gtc clips having some in-effect rendered in: at the start of the production, the texture starts to play from its beginning, continuing to the loop over its trailing part.

PlayIn_Pause_PlayOut



Starts playback of texture in sync with the start of the production, pausing playback of the gtc clip at the frame specified by `PauseFrame` property value. As the production continues close to its end, the playback is unpaused and the trailing part of the cursor clip is played out, finishing together with the end of the production.

Column

`chyronhego.paint.tools.column.ColumnTool`

Displays a textured 3D column with a textured spot on the ground. Supports keyframed animation. Requires pitch calibration.

Tracking: 3D
Keyframing: yes
Chroma key: column spot only

Counter

`aki.paint.tools.counter.CounterTool`

Displays a numeric value over background graphics. In the specified time interval, the value runs in given range.

Two modes are supported (see CountMode property):

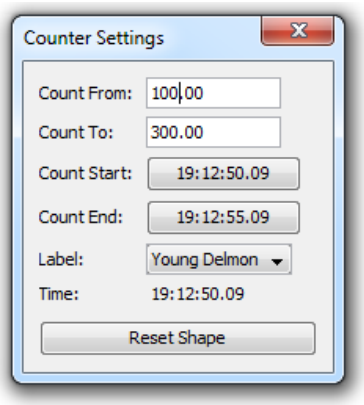
FROM_TO	Counter shows values from a CountFrom to a CountTo range, starting at Count Start time and finishing at Count End time.
SECONDS	Counter shows the number of seconds from Count Start time to Count End time.

Counting in pause can be enabled by setting the RelativeCounting property to true. When set to false, the counter shows the same number during pause.

Optionally, an additional label can be displayed, showing a text content retrieved from accompanying labels.csv file. Individual labels in file are newline-separated. To split label text into multiple lines, use a comma character.

Placing the counter shows up a counter settings dialog:

Count From	Sets a counter starting value (in FROM_TO count mode).
Count To	Sets counter final value (in FROM_TO count mode).
Count Start	Pressing the button sets current video time as a count start time.
Count End	Pressing the button sets current video time as a count end time.
Label	If the label is enabled, user can choose between possible label texts found in a labels.csv file.
Time	Current video time.
Reset Shape	Press button to reset shape of the counter graphics while keeping its position. Helps to get rid of distortion produced by tracking.



After closing, the dialog can be opened again by RMB over the tool button, selecting the Edit... item in the popup menu. Placing another counter into the video also makes the dialog appear.

Parameters “Count From” and “Count To” could be remotely modified by the following XML-based protocol actions:

```
<SetToolCounterFrom value="..."/>
<SetToolCounterTo value="..."/>
```

See section Remote Control for more details.

Tracking: Screen, Motion
Keyframing: no
Chroma key: no

Configurable parameters

`CountMode`	Decides counting range: either the <CountFrom, CountTo> (the FROM_TO CountMode) or a number of seconds since CountStart time (the SECONDS mode)
`File`	Name of the background graphics resource file (image/clip).
`Size`	Float value defining size of the background graphics. Value of 1.0 means that the background pixels have the same height as the background video pixels.
`BackgroundPreroll`	Specifies difference between background and counter text production beginning, in video fields.
`ProductionEndDelay`	When enabled specifies how long the counter is in production (including out effect) after reaching the Count End time, in video fields. Production is infinite when disabled.
`LabelEnabled`	If true, additional label text is displayed, according to the Label... properties settings. If false, only the counter text is shown.
`LabelTextValue`	Label text content.
`Label<FontParameter>`	These parameters define the font style of the label text. See font parameters below.
`TextFormat`	Formatting rules for counter text. The place holder for the value is `"%f"`. To specify, for example, 3 decimal places use `"%.3f"`. You can add any text before/after the number. To specify, for example, units use `"%.1f sec"`.
`AnchorPoint`	Reference point of the counter graphics. Values are in the <0,1> range. [0.5,0.5] by default.
`<FontParameter>`	These parameters define the font style of the counter text. See font parameters below.

Font parameters

`TextPosition`	Text position relative to the whole composition. Value is vector [x, y] in pixels of the background graphics. Value [0,0] is left/top corner.
`TextSize`	Height of one text line in pixels of the background graphics.
`FontHAlign`	Horizontal alignment { LEFT | CENTER | RIGHT | DOT_FIRST | DOT_LAST }
`FontHAlignChar`	Horizontal alignment character
`FontVAlign`	Vertical alignment { TOP | CENTER | BOTTOM | FIRST | LAST }

FontLineSpacing	Relative line spacing. Value 1.0 means spacing by the height of the font.
FontLineFlowPolicy	Behavior for lines wider then defined limit. { DONT_CARE | SCALE | UNIFORM_SCALE | SOFT_NEWLINE | HARD_NEWLINE | SOFT_SKIP | HARD_SKIP | SINGLE_LINE }
FontLineWidth	Line width limit. (see FontLineFlowPolicy)
FontFixedPitch	Fixed character pitch. (disables kerning defined by font).
FontKerning	Additional kerning.
FontCapitalsRatio	If enabled (value > 0) then all characters are made capitals. The original non capital letters are rendered smaller. This value defines factor for that scale.
FontName	Name of the font. All system fonts are available.
FontResolution	Font texture resolution in pixels. It should approximately match the pixel size on the screen.
FontBaseColorMode	Coloring text mode. { SOLID | CHAR_4POINTS | LINE_4POINTS | GLOBAL_4POINTS | LINE_2GRADIENTS }
FontBaseColor	Color to be used when mode is SOLID
FontBaseColorP1	Color for gradient coloring
FontBaseColorP2	Color for gradient coloring
FontBaseColorP3	Color for gradient coloring
FontBaseColorP4	Color for gradient coloring
FontOutlineSize	Size of outline in pixels. Set to zero to disable outline.
FontOutlineColor	Outline color.
FontShadowOffset	Relative offset of shadow to normal text.
FontShadowBlur	Amount of blur of the shadow.
FontShadowColor	Shadow color.

Cursor

aki.paint.tools.cursor.CursorTool

Image or .gtc clip that is put into image by clicking a mouse. Can be dragged with a mouse after it is placed. Supports keyframe animation.

File	Name of the cursor resource file (image/clip).
Perspective	Amount of artificial perspective distortion applied to the cursor graphics. Value between 0.0 (no perspective) and 1.0 (flat)

Size	Float value defining size of the cursor. Value of 1.0 means that the cursor pixels have the same height as the background video pixels.
ColorMatrix	*Advanced cursor coloring. Changed from GUI.*
Opacity	Opacity multiplication constant.
PivotFrame	Index of the gtc clip frame used to create thumbnail. Ignored if the cursor is image-based. Ignored if Thumbnail property defines thumbnail image file.
SizeReference	Reference frame for the Size value. PREVIEW – if Size value equals to 1.0, cursor resource pixels match y-pixels of the preview. VIDEO – if Size value equals to 1.0, cursor resource pixels match y-pixels of the video.
SizeTracking	Controls how graphics size is affected by background video zoom changes. KEEP_CONSTANT - size stays constant when background zooms in/out FOLLOW_ZOOM - size follows background zoom change
AnchorPoint	Reference point of the cursor graphics. Values are in the <0,1> range. [0.5,0.5] by default.
AutoPause	If enabled, pause of AutoPauseDuration length is created.
AutoPauseDuration	Duration of automatically generated pause.
3DScale	By enabling the value, the graphics is placed on 3D pitch, when available. Scale value defines default size of the graphics in 3D space.

Tracking: Screen, ScreenMotion, 3D
Keyframing: yes
Chroma key: yes

Example

```
<Tool type="aki.paint.tools.cursor.CursorTool">
   <Id value="CursorCircle"/>
   <File value="circle.png"/>
   <Size value="0.5"/>
  <Perspective value="0"/>
   <KeyframeAnimEnabled value="1" />
</Tool>
```

Redefining GTC cursor color with ColorMatrix property

Paint allows you to customize the color of .gtc clip-cursor. We provide animcursor1.gtc. You can redefine the colors in that cursor.

Use the popup menu on a tool button to open the Properties editor. Change color mode of the ColorMatrix property to Three Colors mode. Colors are then replaced by following rules:

- The first color replaces Red
- The second color replaces Green
- The third color replaces Blue

- The final color is computed as a mix of all three replacing colors modulated by the original RGB values in the clip.

If the provided cursor has 2 colors only, select 2 colors of your choice and the original colors in the cursor will be replaced.

Cursor With Trail

`aki.paint.tools.advancedcursor.CursorWithTrailTool`

Image or .gtc clip that is put into image by clicking a mouse. Can be dragged with a mouse after it is placed. Supports keyframe animation. While the keyframe is animated, it leaves a trail stroke. Trail stroke can have applied fade off gradient (see `TrailAlphaGrad...` properties).

Used preferably with a `PlaceModelScreenMotion` place.

To not show trailing stroke through the cursor, usage of masked .gtc clip for cursor is recommended. See Masked .gtc clip chapter for details.

`Size`	Float value defining size of the cursor. Value of 1.0 means that the cursor pixels have the same height as the background video pixels.
`SizeTracking`	Controls how cursor size is affected by background zooming. KEEP_CONSTANT – Cursor size stays constant when background zooms in/out. Default. FOLLOW_ZOOM - Cursor size follows background zoom changes.
`Perspective`	Amount of artificial perspective distortion applied to the cursor graphics. Value between 0.0 (no perspective) and 1.0 (flat)
`StrokeColor`	Coloring of the trail stroke.
`ColorMatrix`	*Advanced cursor coloring. Changed from GUI.*
`Opacity`	Opacity multiplication constant. Applied on the cursor itself only.
`TrailAlphaGradStart`	Distance from the head of the trail where the alpha gradient effect starts.
`TrailAlphaGradEnd`	Distance from the head of the trail where the alpha gradient effect ends.
`TrailAlphaGradPower`	Exponent to control alpha gradient shape. Value 1..N. For example value 1.0 makes the gradient linear.
`TrailAlphaGradEnabled`	Enables or disables trail alpha gradient effect.
`TrailType`	Type of trail. Possible values are: FROM_START – Trail drawn from start of the path to cursor. Default. TO_END – Trail drawn from cursor to end of the path. START_TO_END – Trail drawn from start to end of the path. Cursor moves along the trail.
`File`	Name of the cursor resource file (image/clip).
`StrokeThickness`	Relative trail stroke thickness. Related to a single pixel of 576i video.
`StrokeStretch`	Range that is stretched along the trail stroke, in relative coordinates (0.0-1.0). Texture areas outside the range are not stretched.
`StrokeTextureFile`	Image/clip file used as a trail stroke texture. Empty value results in solid color stroke.
`StrokeTexturePauseFrame`	Index of the texture frame to pause playback at. Applied when `PlayIn_Pause_PlayOut` PlayMode is set. Ignored for image-based textures.
`StrokeTexturePlayMode`	Decides how the .gtc clip is played on production. One of `Loop`, `PlayIn_Loop`, `PlayIn_Pause_PlayOut` values. See GTC Texture Play Modes for details.
`StrokeTextureSpeed`	Speed of possible texture movement along the stroke.
`3DScale`	By enabling the value, the graphics is placed on 3D pitch, when available. Scale value defines default size of the graphics in 3D space.

Tracking: ScreenMotion, 3D
Keyframing: yes
Chroma key: yes

Example

```
<Tool type="aki.paint.tools.advancedcursor.CursorWithTrailTool">
  <Id value="CursorWithTrail"/>
  <File value="playerMasked.gtc"/>
  <Size value="1.2"/>
  <Thumbnail value="tCursorWithTrail.png" />
  <ThumbnailColoringEnabled value="true" />
  <ChromaKeyEnabled value="1" />
  <KeyframeAnimEnabled value="1"/>
  <PlaceType
value="aki.productplacement.place.screen.PlaceModelScreenMotion"/>
  <StrokeColor value="[0.0, 1.0, 0.0, 1.0]"/>
  <StrokeThickness value="2"/>
  <StrokeTexture value="line_dash.png"/>
</Tool>
```

CutOut

`aki.paint.cutout.tools.cutout.CutOutTool`

CutOut tool allows to highlight, move or resize players.

Object detection

After selecting the tool, Paint starts to automatically detect suitable objects in video. Detected objects are presented to user by displaying their bounding boxes. Because the automatic detection is based on chroma keyer data, it is necessary to have chroma keyer set properly to get suitable detection results.

Mask editor

Results of automatic detection can be further enhanced using mask editor. To start the mask editor, click a scissors button in a preview, then click an object you want to edit. Preview window zooms to the selected object and a mask editor toolbar shows up.

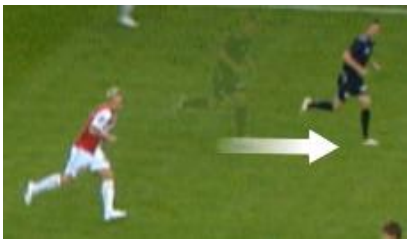
Mask editor supports multiple different tools:

	Brush tool. Fine-tuning of object shape. By default, stroke removes object area. Holding Ctrl key inverts the effect, so the stroke adds an object area. Size of the brush is controlled by a mouse wheel.
	Knife tool. Quick removal of large object parts using cutting line. Clicking into the image creates a cutting line that should split object into two or more separate areas. When finished, hold the Shift key and click the object area(s) to remove from the object.
	Lasso tool. Quick removal of closed object areas. Clicking into the image creates a closed polygon that splits object into two separate areas. When finished, click the polygon inner area to remove all parts of the object that are not inside the polygon. Click outside the polygon to remove the inner part.
	Revert back tool. Reverts the object to the state when the editor was opened.
	When finished, leave the mask editor with this button.

Effects

The tool is applied by clicking one of the players. CutOut effects are applied immediately after the player has been selected.

CutOut tool supports three effects that can be combined arbitrarily.

Glow		Adds a color glow effect around the player.
Move		Moves the player to a different position by dragging him with a mouse. The original location is patched with surrounding background texture. Optionally, a 'ghost' player can be rendered there. It is possible to show an arrow connector between the original and moved position.
Scale		Scales player using mouse wheel while holding Ctrl key.

Preparing for production

It is recommended to use the CutOut tool in Playback Speed tool pause, otherwise it will be only applied to the current frame. When created, CutOut effects are applied from the time of the creation for a time period defined by the *DefaultProductionLength* property. To control duration of CutOut effects production, use the Production Start/End button.
All types of In/Out Effects are supported, though mixing DRAW with multiple keyframe move effect is not recommended.

Keyframed move effect
Since Paint 9.4 it is possible to create a keyframed animation of move effect, meaning the player will smoothly pass through any path defined by these keyframes.

1. Jog to the timecode when the player should start moving (in a pause).
2. Click on the desired player.

3. Jog forward to another time.
4. Drag the player to another place.

To define more keyframes, simply repeat steps 3 and 4. If a keyframe is defined outside current CutOut production, it is automatically extended. The move effect will always start in the original player position so it's not possible to drag the player in the timecode of creation.

Tool properties reference

`DefaultProductionLength`	Initial length of production after the tool is created.
`ShowBoundingBoxes`	Show bounding boxes of the Cutout drawings even when Operator mode is off.
`Glow/Blur`	Amount of blur applied to the glow effect. Influences glow size.
`Glow/Color`	Glow effect color.
`Glow/Intensity`	Intensity of glow effect.
`Move/Connector`	Connector stroke style. Fetches styles of freehand tools present in a toolbar. Set to -none- to not use connector stroke.
`Move/ConnectorColor`	Color of the connector stroke.
`Move/GhostOpacity`	Opacity of the player ghost rendered at the original player position. 0 by default.
`Move/PlayerOpacity`	Opacity of the moved player. 1 by default.
`Scale/ScaleCentre`	Pivot point of the scale effect, in relative mask coordinates. E.g.: [0,0] uses mask bottom-left corner as a scale pivot. [0.5,0.5] uses mask center as a scale pivot.

Sequential Cursor

`aki.paint.tools.cursor.SequentialCursorTool`

The tool has predefined sequence of images/gtc clips. Once you place sequential cursor into the screen, it automatically changes its resource file. Next placed cursor is different.
This is useful e.g. to place sequence of cursors with numbers. The number is automatically increased.
The sequence is reset on Clear or after reaching the maximum index within the sequence.

`FilePattern`	Pattern of the cursor resource files (image/clip). Example: **`numbered_cursor_red_%s.png`** The '%s' substring will be replaced by numbers from the range defined in 'SequenceBegin' and 'SequenceEnd' elements.
`SequenceBegin`	Begin index of the sequence of files
`SequenceEnd`	End index of the sequence of files
`Perspective`	Amount of artificial perspective distortion applied to the cursor graphics. Value between 0.0 (no perspective) and 1.0 (flat)
`Size`	Float value defining size of the cursor. Value of 1.0 means that the cursor pixels have the same height as the background video pixels.
`ColorMatrix`	*Advanced cursor coloring. Changed from GUI.*
`Opacity`	Opacity multiplication constant.
`PivotFrame`	Index of the gtc clip frame used to create thumbnail. Ignored if the cursor is image-based. Ignored if Thumbnail property defines thumbnail image file.
`SizeReference`	Reference frame for the Size value. PREVIEW – if Size value equals to 1.0, cursor resource pixels match y-pixels of the preview. VIDEO – if Size value equals to 1.0, cursor resource pixels match y-pixels of the video.
`3DScale`	By enabling the value, the graphics is placed on 3D pitch, when available. Scale value defines default size of the graphics in 3D space.

Tracking: Screen, ScreenMotion, 3D
Keyframing: yes
Chroma key: yes

Example

```
<Tool type="aki.paint.tools.cursor.SequentialCursorTool">
  <Id value="SequentialCursor"/>
  <FilePattern value="numbered_cursor_red_%s.png"/>
  <Size value="1.2"/>
  <ChromaKeyEnabled value="1" />
  <PlaceType
  value="aki.productplacement.place.screen.PlaceModelScreenMotion"/>
  <StrokeFile value="lineDashed.fhs" />
  <KeyframeAnimEnabled value="1"/>
  <SequenceBegin value="1"/>
  <SequenceEnd value="9"/>
</Tool>
```

Number tool

`aki.paint.tools.numbertool.NumberTool`

Similar to sequential cursor, a number value is rendered as pure text.
Value is incremented every time, when a drawing is placed on the screen.
The value is set to first number value on Clear or after reaching the last number value.

`FirstNumberValue`	First value of sequence. Set to 1 when disabled.
`LastNumberValue`	Last value of sequence. When disabled actual value is never reset to first value.
`Size`	Float value defining size of the cursor. Value of 1.0 means that the cursor pixels have the same height as the background video pixels.
`ColorMatrix`	*Advanced cursor coloring. Changed from GUI.*
`3DScale`	By enabling the value, the graphics is placed on 3D pitch, when available. Scale value defines default size of the graphics in 3D space.

Font parameters

`TextPosition`	Offset of the text layer line relative to reference point, in background graphics pixels.
`TextPositionReference`	Reference point for the text position offset. Can be TOP_LEFT(default) or TOP_RIGHT of the background graphics.
`TextSize`	Height of one text line in pixels of the background graphics.
`FontHAlign`	Horizontal alignment { LEFT | CENTER | RIGHT | DOT_FIRST | DOT_LAST }
`FontHAlignChar`	Horizontal alignment character
`FontVAlign`	Vertical alignment { TOP | CENTER | BOTTOM | FIRST | LAST }
`FontFixedPitch`	Fixed character pitch. (disables kerning defined by font).
`FontKerning`	Additional kerning.
`FontCapitalsRatio`	If enabled (value > 0) then all characters are made capitals. The original non capital letters are rendered smaller. This value defines factor for that scale.
`FontName`	Name of the font. All system fonts are available.
`FontResolution`	Font texture resolution in pixels. It should approximately match the pixel size on the screen.
`FontBaseColorMode`	Coloring text mode. { SOLID | CHAR_4POINTS | LINE_4POINTS | GLOBAL_4POINTS | LINE_2GRADIENTS }
`FontBaseColor`	Color to be used when mode is SOLID
`FontBaseColorP1`	Color for gradient coloring
`FontBaseColorP2`	Color for gradient coloring

FontBaseColorP3	Color for gradient coloring
FontBaseColorP4	Color for gradient coloring
FontOutlineSize	Size of outline in pixels. Set to zero to disable outline.
FontOutlineColor	Outline color.
FontShadowOffset	Relative offset of shadow to normal text.
FontShadowBlur	Amount of blur of the shadow.
FontShadowColor	Shadow color.

Tracking: Screen, ScreenMotion, 3D
Keyframing: yes
Chroma key: no

Example

```
<Tool type="aki.paint.tools.numbertool.NumberTool">
  <Thumbnail value="tNumberTool.png"/>
  <File value="background.png"/>
  <KeyframeAnimEnabled value="true"/>
  <Size value="0.5"/>
  <FontHAlign value="CENTER"/>
  <FontVAlign value="CENTER"/>
  <FontResolution value="50"/>
  <TextSize value="100"/>
  <ThumbnailColoringEnabled value="false"/>
</Tool>
```

Linked Cursors

`aki.paint.tools.advancedcursor.LinkedCursorsTool`

Group of cursors connected by stroke link. Defined by subsequent clicking into the video. Cursors can be dragged to create keyframe animation.
To not show link stroke through the cursors, usage of masked .gtc clip for cursors is recommended. See Masked .gtc clip chapter for details.
Linked cursors

`Closed`	True – cursors are linked all around. False – cursors are linked from the first to the last.
`FillEnabled`	Enables fill.
`FillTextureFile`	Name of the fill texture resource file (image/clip).
`FillColor`	Coloring of the fill. Used only when Closed is True.
`Size`	Float value defining size of the cursor. Value of 1.0 means that the cursor pixels have the same height as the background video pixels.
`ColorMatrix`	*Advanced coloring of cursor. Changed from GUI.*
`Opacity`	Opacity multiplication constant. Applied on the cursors only.
`Perspective`	Amount of artificial perspective distortion applied to the cursor graphics. Value between 0.0 (no perspective) and 1.0 (flat)
`StrokeThickness`	Relative stroke thickness. Related to a single pixel of 576i video.
`StrokeSmooth`	If true, the stroke data are being smoothed to remove possible input jittering.
`StrokeSmoothRadius`	Affects smoothing effect strength. Value 4 leads to acceptable results.
`StrokeLineJoin`	Joining of stroke segments. One of: [ROUND | MITER | BEVEL]
`StrokeLineCap`	Stroke line endpoint shape. One of: [ROUND | FLAT]. ROUND by default.
`StrokeAlphaGradient`	Applies alpha gradient along the stroke length. 4-number vector, in a [t0, t1, a0, a1] form, where t0, t1 are relative coordinates from <0,1> range and a0, a1 are alpha values in t0 and t1, respectively.
`StrokeStretch`	Range that is stretched along the stroke, in relative coordinates (0.0-1.0). Texture areas outside the range are not stretched.
`StrokeTexture`	Image file used as a connection stroke texture. Empty value results in solid color stroke.
`StrokeTextureSpeed`	Speed of possible texture movement along the stroke.
`StrokeColor`	Coloring of the polygon stroke.
`3DScale`	By enabling the value, the graphics is placed on 3D pitch, when available. Scale value defines default size of the graphics in 3D space.

It is possible to display a text showing a distance between cursor points. Distance text appearance is controlled from the Distance property tab. An important property is a **TrackingType** that decides how the distance text is presented:

NONE	On-screen distance text. Overall length of the arrow is shown. TextPosition defines the absolute position of the text, in normalized screen coordinates.
SHAPE_ALIGNED	Distance text in pitch plane. Shown for each arrow segment. TextPosition defines metric offset of text from the center point of linked cursors segment.
CAMERA_FACING	3D distance text standing on pitch. Shown for each arrow segment. TextPosition defines metric offset of text from the center point of linked cursors segment.

Distance text supports background texture. Background texture is controlled from the Distance Texture properties tab.

Tracking: Screen, ScreenMotion, 3D
Keyframing: yes
Chroma key: yes

Example:

```
<Tool type="aki.paint.tools.advancedcursor.LinkedCursorsTool">
   <Id value="LinkedCursors"/>
   <File value="playerMasked.gtc"/>
   <Size value="1.2"/>
   <FillEnabled value="true"/>
   <FillColor value="[1.0,0.0,0.0,1.0]"/>
   <Thumbnail value="tLinkedCursors.png"/>
   <ThumbnailColoringEnabled value="true"/>
   <ChromaKeyEnabled value="1"/>
   <KeyframeAnimEnabled value="1"/>
   <PlaceType value="aki.productplacement.place.screen.PlaceModelScreen"/>
   <StrokeThickness value="3"/>
   <StrokeColor value="[0.0, 1.0, 0.0, 1.0]"/>
</Tool>
```

Freehand

`aki.paint.tools.freehand.FreehandTool`

Allows to paint strokes by dragging the mouse over the image.

`Thickness`	Relative stroke thickness. Related to a single pixel of 576i video.
`Smooth`	Enables stroke smoothing
`SmoothRadius`	Radius of stroke smoothing algorithm
`LineJoin`	Joining of stroke segments. One of: [ROUND | MITER | BEVEL]
`LineCap`	Line endpoint shape. One of: [ROUND | FLAT]. ROUND by default.
`AlphaGradient`	Applies alpha gradient along the stroke length. 4-number vector, in a [t0, t1, a0, a1] form, where t0, t1 are relative coordinates from <0,1> range and a0, a1 are alpha values in t0 and t1, respectively.
`Stretch`	Range that is stretched along the stroke, in relative coordinates <0,1>. Texture areas outside the range are not stretched.
`Texture`	Image/clip file used as a stroke texture. Empty value results in solid color stroke.
`TextureSpeed`	Speed of possible texture movement along the stroke.

Tracking: Screen, ScreenMotion
Keyframing: no
Chroma key: yes

```
<Tool type="aki.paint.tools.freehand.FreehandTool">
   <Id value="Freehand" />
   <Color value="[1.0, 1.0, 1.0, 1.0]" />
   <Thumbnail value="tFreehand.png" />
   <ThumbnailColoringEnabled value="true" />
   <ChromaKeyEnabled value="false"/>
   <Thickness value="10"/>
   <Smooth value="true"/>
   <LineJoin value="MITER"/>
   <Texture value="arrow.png"/>
   <TextureSpeed value="1"/>
</Tool>
```

Line

`aki.paint.tools.freehand.LineTool`

Allows to paint straight line strokes by dragging mouse over the image from the line's start point to its end point.

`Thickness`	Relative stroke thickness. Related to a single pixel of 576i video.
`LineCap`	Line endpoint shape. One of: [ROUND | FLAT]. ROUND by default.
`AlphaGradient`	Applies alpha gradient along the stroke length. 4-number vector, in a [t0, t1, a0, a1] form, where t0, t1 are relative coordinates from <0,1> range and a0, a1 are alpha values in t0 and t1, respectively.
`Stretch`	Range that is stretched along the stroke, in relative coordinates <0,1>. Texture areas outside the range are not stretched.
`Texture`	Image/clip file used as a stroke texture. Empty value results in solid color stroke.
`TextureSpeed`	Speed of possible texture movement along the stroke.

Tracking: Screen, ScreenMotion
Keyframing: no
Chroma key: yes

```
<Tool type="aki.paint.tools.freehand.LineTool">
   <Id value="StraightArrow" />
   <Color value="[1.0, 1.0, 1.0, 1.0]" />
   <Thumbnail value="tStraightArrow.png" />
   <ThumbnailColoringEnabled value="true" />
   <PlaceType
value="aki.productplacement.place.screen.PlaceModelScreenMotion" />
    <Thickness value="10"/>
    <Stretch value="[0.05, 0.08]"/>
    <Texture value="StraightArrow.png"/>
    <LineCap value="FLAT"/>
</Tool>
```

Animated Line

`aki.paint.tools.animatedline.AnimatedLineTool`

Allows to paint straight line stroke by creating begin and end point. Points can be keyframe animated.

`Color`	Fill color. Only if the texture file is empty.
`Thickness`	Relative stroke thickness. Related to a single pixel of 576i video.
`LineCap`	Line endpoint shape. One of: [ROUND | FLAT]. ROUND by default.
`AlphaGradient`	Applies alpha gradient along the stroke length. 4-number vector, in a [t0, t1, a0, a1] form, where t0, t1 are relative coordinates from <0,1> range and a0, a1 are alpha values in t0 and t1, respectively.
`Stretch`	Range that is stretched along the stroke, in relative coordinates <0,1>. Texture areas outside the range are not stretched.
`Texture`	Image/clip file used as a stroke texture. Empty value results in solid color stroke.
`TextureSpeed`	Speed of possible texture movement along the stroke.

Tracking: Screen, ScreenMotion
Keyframing: yes
Chroma key: yes

```
<Tool type="aki.paint.tools.animatedline.AnimatedLineTool">
  <Thumbnail value="tAnimatedLine.png"/>
  <ThumbnailColoringEnabled value="true"/>
</Tool>
```

Polygon

`aki.paint.tools.freehand.PolygonTool`

Allows to define closed polygon by subsequently clicking points into the image. Polygon creation is finished by either clicking back to the starting point or by switching to different tool.

`Thickness`	Relative stroke thickness. Related to a single pixel of 576i video.
`LineJoin`	Joining of stroke segments. One of: [ROUND | MITER | BEVEL]
`AlphaGradient`	Applies alpha gradient along the stroke length. 4-number vector, in a [t0, t1, a0, a1] form, where t0, t1 are relative coordinates from <0,1> range and a0, a1 are alpha values in t0 and t1, respectively.
`Stretch`	Range that is stretched along the stroke, in relative coordinates <0,1>. Texture areas outside the range are not stretched.
`Texture`	Image/clip file used as an outline stroke texture. Empty value results in solid color stroke.
`TextureSpeed`	Speed of possible texture movement along the stroke.
`TextureLoopStart`	Frame that the texture animation wraps to. Applied only if a gtc clip is applied as a stroke texture.
`ShowFill`	True – the internal area of the polygon is filled with texture/color. False – internal area of the polygon is empty.
`FillTexture`	Texture applied to the polygon internal area.
`FillColor`	Coloring of the polygon internal area. If FillTexture is defined, the FillColor colorizes texture.

Tracking: Screen, ScreenMotion
Keyframing: no
Chroma key: yes

```
<Tool type="aki.paint.tools.freehand.PolygonTool">
   <Id value="Polygon"/>
   <Color value="[1.0, 1.0, 1.0, 1.0]"/>
   <Thumbnail value="tPolygon.png"/>
   <ThumbnailColoringEnabled value="true"/>
   <PlaceType
   value="aki.productplacement.place.screen.PlaceModelScreenMotion"/>
   <ChromaKeyEnabled value="1"/>
   <FillTexture value="white.png"/>
   <FillColor value="[0.0,0.0,0.5,0.5]"/>
   <Thickness value="5"/>
    <LineCap value="FLAT"/>
</Tool>
```

Trail

`aki.paint.tools.trailtool.TrailTool`

Keyframe animated trail stroke.
Trail stroke can have applied fade off gradient (see `TrailAlphaGrad...` properties).
Used preferably with a `PlaceModelScreenMotion` place.

`TrailAlphaGradEnabled`	Enables or disables trail alpha gradient effect.
`TrailAlphaGradLength`	Length of the alpha gradient effect, in screen normalized coordinates along stroke.
`TrailAlphaGradPower`	Exponent to control alpha gradient shape. Value 1..N. For example value 1.0 makes the gradient linear.
`TrailAlphaGradStart`	Distance from the head of the trail where the alpha gradient effect starts.
`LineCap`	Line endpoint shape. One of: [ROUND | FLAT]. ROUND by default.
`LineJoin`	Joining of stroke segments. One of: [ROUND | MITER | BEVEL]
`Smooth`	Enables stroke smoothing
`SmoothRadius`	Radius of stroke smoothing algorithm
`Stretch`	Range that is stretched along the stroke, in relative coordinates <0,1>. Texture areas outside the range are not stretched.
`Texture`	Image/clip file used as a stroke texture.
`TextureSpeed`	Speed of possible texture movement along the stroke.
`Thickness`	Relative stroke thickness. Related to a single pixel of 576i video.
`Type`	Type of trail. Possible values are: FROM_START – Trail drawn from start of the path to cursor. Default. TO_END – Trail drawn from cursor to end of the path. START_TO_END – Trail drawn from start to end of the path. Cursor moves along the trail.

Tracking: Screen, ScreenMotion
Keyframing: yes
Chroma key: yes

```
<Tool type="aki.paint.tools.trailtool.TrailTool">
    <ResourcePath value="Trail"/>
    <Thumbnail value="tTrail.png"/>
    <Color value="[1.0, 1.0, 1.0, 1.0]"/>
    <Texture value="trail.png"/>
    <Stretch value="[0.1,0.5]"/>
</Tool>
```

Zone Polygon

`aki.paint.tools.advancedcursor.DecoratedPolygonTool`

Filled polygon area which points are defined by clicking into the image. Displays image/clip decorations at the polygon vertices.

`FillTexture`	Name of the polygon fill texture resource file (image/clip)
`FillColor`	Coloring of the polygon fill.
`File`	Name of the vertex decoration resource file (image/clip).
`Size`	Float value defining size of the vertex decoration. Value of 1.0 means that the decoration pixels have the same height as the background video pixels.
`StrokeThickness`	Relative polygon stroke thickness. Related to a single pixel of 576i video.
`StrokeStretch`	Range that is stretched along the stroke, in relative coordinates (0.0-1.0). Texture areas outside the range are not stretched.
`StrokeTexture`	Image file used as an outline stroke texture. Empty value results in solid color stroke.
`StrokeTextureSpeed`	Speed of possible texture movement along the stroke.
`StrokeColor`	Coloring of the polygon stroke.
`3DScale`	By enabling the value, the graphics is placed on 3D pitch, when available. Scale value defines default size of the graphics in 3D space.

Tracking: Screen, ScreenMotion, 3D
Keyframing: yes
Chroma key: yes

Example:

```
<Tool type="aki.paint.tools.advancedcursor.DecoratedPolygonTool">
   <Id value="DecoratedPolygon"/>
   <File value="spark.png"/>
   <Size value="0.3"/>
   <Thumbnail value="tDecoratedPolygon.png"/>
   <ThumbnailColoringEnabled value="true"/>
   <ChromaKeyEnabled value="1"/>
   <PlaceType
   value="aki.productplacement.place.screen.PlaceModelScreenMotion"/>
   <StrokeThickness value="2"/>
  <StrokeTexture value="stroke.png"/>
  <StrokeStretch value="[0, 1]"/>
   <StrokeColor value="[1.0, 1.0, 1.0, 1.0]"/>
   <FillTexture value="white.png"/>
   <FillColor value="[0.0,0.0,0.5,0.5]"/>
   <Closed value="true"/>
</Tool>
```

Thumbnail coloring

Thumbnail coloring is controlled by ThumbnailColoringEnabled property. When is enabled, thumbnail pixels are colored with tool colors according to following table:

RED	FillColor
GREEN	StrokeColor
BLUE	Not used. Always white.

Label

`aki.paint.tools.label.LabelTool`

Label tool combines an on-screen image/clip displaying some information with a keyframe-animated pointer line.
The tool allows to choose from multiple label contents: any image/.gtc clip found in a `LabelContentPath` folder is considered to be a label content option. Currently selected option file name is displayed over the tool thumbnail.
The tool supports optional text layer containing one or more texts.

`LabelFile`	Name of the label resource file (image/clip)
`ConnectorAnchor`	Connector start point related to a label position. Anchor is scaled by Size property.
`Size`	Float value defining size of the label. Value of 1.0 means that the label pixels have the same height as the background video pixels.
`ConnectorThickness`	Relative connector thickness. Related to a single pixel of 576i video.
`ConnectorTexture`	Image/clip file used as a connector texture.
`ConnectorStretch`	Range that is stretched along the connector, in relative coordinates <0,1>. Texture areas outside the range are not stretched.
`ThumbnailTextYPos`	Y-position of the thumbnail text position, in pixels
`ThumbnailTextSize`	Size of the thumbnail text
`ThumbnailText`	Text rendered into the thumbnail
`ThumbnailTextColor`	Color of the thumbnail text in rgba floats notation. Black by default.
`LabelContentPath`	Path to a label content resources

Tracking: Screen,
Keyframing: yes
Chroma key: no

Example

```
<Tool type="aki.paint.tools.label.LabelTool">
   <Thumbnail value="tLabel.png"/>
   <LabelFile value="../DriverAssets/driver1.png"/>
   <ConnectorThickness value="5" />
   <ConnectorTexture value="connector.gtc" />
   <ConnectorTextureLoopStart value="49" />
   <ConnectorStretch value="[0, 0.99]" />
   <LabelLoopStart value="350"/>
   <ThumbnailTextYPos value="30"/>
   <ThumbnailTextSize value="15"/>
   <ThumbnailText value="Label"/>
   <ThumbnailTextColor value="[0,0.5,0,1]"/>
   <LabelContentPath value="../DriverAssets"/>
</Tool>
```

Light Shaft

`aki.paint.tools.offside.LightShaftTool`

Light shaft tool allows to highlight a player by spotlight. Tool is composed from a spot at the bottom and a shaft coming from the top.

`Perspective`	Amount of artificial perspective distortion applied to the spot graphics. Value between 0.0 (no perspective) and 1.0 (flat)
`Color`	Color of the spot and the shaft.
`File`	Spot file. Image or a .gtc clip.
`ShaftFile`	Shaft file. Image or a .gtc clip.
`ShaftType`	• FIXED – shaft origin is fixed on position defined by ShaftOriginX and ShaftOriginY. • RELATIVE – shaft origin is defined by spot position + ShaftOriginX and ShaftOriginY. • NONE – no shaft is shown.
`ShaftOriginX`	Horizontal shaft origin position.
`ShaftOriginY`	Vertical shaft origin position.
`Size`	Size of spot.
`TopShaftSize`	Top shaft size ratio to the spot size.
`BottomShaftSize`	Bottom shaft size ratio to the spot size.
`BottomShaftOffset`	Offset of the spot from the bottom of the shaft. Relative to the spot size.

Tracking: no
Keyframing: yes
Chroma key: yes

Example

```
<Tool type="aki.paint.tools.lightshaft.LightShaftTool">
    <File value="Spot.png"/>
    <ShaftFile value="Shaft.gtc"/>
    <Size value="0.6"/>
    <Perspective value="0.17230769"/>
    <Color value="[1.0,0.9,0.793,1.0]"/>
    <ShaftType value="FIXED"/>
    <ShaftOriginX value="0.44"/>
    <ShaftOriginY value="1.5"/>
    <TopShaftSize value="0.2"/>
    <BottomShaftSize value="0.85"/>
</Tool>
```

Eye View

`aki.paint.tools.eyeview.EyeViewTool`

Eye view tool allows to highlight the view range from the player's point of view.

`File`	Name of the texture resource file (image/clip).
`Color`	Color of the texture.
`LoopStart`	Applied on gtc texture. Specifies frame from which the playback loops. 0 by default.
`PauseFrame`	Applied on gtc texture. Specifies frame on which the playback is paused. -1 by default (no pause).

Tracking: no
Keyframing: yes
Chroma key: yes

Example

```
<Tool type="aki.paint.tools.eyeview.EyeViewTool">
    <File value="EyeView.png"/>
    <Thumbnail value="tEyeView.png"/>
    <ThumbnailColoringEnabled value="false"/>
</Tool>
```

Offside

`aki.paint.tools.offside.OffsideTool`

Offside tool allows to show virtual offside at specified time.

After setting up line, and saving to the clip, clip playback automatically stops at the offside time and the line shows up. Starting the playback again starts out effect.

Single offside line can be drawn at one time.

`Texture`	Offside zone texture. Either an image or a .gtc clip.
`TexturePivotFrame`	Pivot frame of the offside zone texture clip. Ignored if an image is used as a texture.
`OffsideZoneWidth`	Pixel width of the offside zone at the bottom of the screen.
`TextureCoordinateSpace`	Pixel dimensions of the offside zone texture.
`TextureOffsideLinePos`	Pixel position of the offside line in offside zone texture.

Tracking: always
Keyframing: no
Keyframing: always

Example

```
<Tool type="aki.paint.tools.offside.OffsideTool">
      <Id value="Offside"/>
      <Texture value="offside_line.png"/>
      <Thumbnail value="tOffside.png"/>
      <OffsideZoneWidth value="500"/>
      <TextureCoordinateSpace value="[512,8]"/>
      <TextureOffsideLinePos value="7"/>
</Tool>
```

Pause

`aki.paint.tools.pause.PauseTool`

Pause tool allows to define point in a clip where the playback automatically stops.

When selected, a pause button and a timeline shows up in preview. Pressing the button puts/removes pause at that time. The pause is indicated by showing a pause mark at the timeline. Pause button turns red at pause.

Pressing the Clear button while the Pause tool is activated removes all pauses from the timeline.

Pauses are saved to the current clip by pressing the Save button.

Pitch Focus

`aki.paint.tools.pitchfocus.PitchFocusTool`

The tool applies discolor and blur effects to an area surrounding pitch bounds. Requires pitch calibration.

Properties of the effect applied to pitch surroundings can be controlled in global Settings dialogue, on the Discolor page.

Configurable parameters

`SoftEdge`	Softening of the effect boundaries

Pitch Zone

`aki.paint.tools.pitchzone.PitchZoneTool`

The tool highlights single or multiple zones on the pitch with desired background texture and border. Zone is a rectangular strip spanning across the shorter dimension of the pitch with configurable length.
Zones can be defined in the Zones section of the tool Property Editor in a simple click and drag manner.

Tracking: 3D
Keyframing: no
Chroma key: yes

Often used zone configurations can be also saved as presets and shared between individual Pitch Zone tools.
Presets are stored in an external file:

`Documents/Paint9/PitchZones/zonePresets.xml`

Configurable parameters

`Thickness`	Relative border thickness.

AlphaGradient	Applies alpha gradient along the stroke length. 4-number vector, in a [t0, t1, a0, a1] form, where t0, t1 are relative coordinates from <0,1> range and a0, a1 are alpha values in t0 and t1, respectively.
Stretch	Range that is stretched along the stroke, in relative coordinates <0,1>. Texture areas outside the range are not stretched.
Texture	Image/clip file used as an outline stroke texture. Empty value results in solid color stroke.
TextureSpeed	Speed of texture movement along the borders.
TextureLoopStart	Frame that the texture animation wraps to. Applied only if a gtc clip is applied as a border texture.
ShowFill	True – the internal area of the zone is filled with texture/color. False – the internal area of the zone is empty.
FillTexture	Texture applied to the zone internal area.
FillColor	Coloring of the zone internal area. If FillTexture is defined, the FillColor colorizes texture.

Trajectory

`aki.paint.tools.trajectory.TrajectoryTool`

Usage

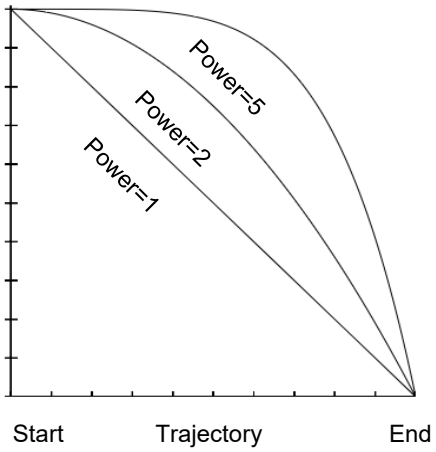
- The tool is designed to work in a clip, not live.
- Jog to the start of the object trajectory and select the tool. Select the flying object and jog forward about half a second (do not make steps more then one second apart). Repeat until the end of the trajectory. You should have about 10 key frames on the trajectory.
- When the ball is lost during the tracking then mark it immediately when it returns back. The tool tries to approximate the missing path.
- During the manual object tracking process, the opacity and thickness gradient is disabled. This is done deliberately to help you see the current shape of the trajectory.
- After you finished making the trajectory save your work to the clip!
- When you select the clip again, you should see the visualized trajectory (including opacity and thickness gradient).
- This tool requires reliable camera motion tracking. The tracking will likely fail if the object is tracked against a solid color area, like a clean blue sky.
- When the object is blurred due to motion blur, try to always select the same spot on the blurred blob. It is better to select the head of the trajectory. Not the average of the blob.

Configurable parameters

`Thickness`	Stroke thickness in pixels in PAL video format.
`Texture`	Image file that is repeated along the trajectory. Empty value results in solid color stroke.
`OpacityStartLength`	Length (in time units) of opacity gradient at the start of the line. The gradient is always from 0 to 1.
`OpacityEndValue`	Opacity value at the end of the curve.
`OpacityEndPower`	Exponent to control gradient shape. Value 1..N. For example value 1.0 makes the gradient linear. See below.
`ThicknessEndFactor`	Thickness factor (to be multiplied with Thickness) at the end of the line.
`ThicknessEndPower`	Exponent to control gradient shape. Value 1..N. For example value 1.0 makes the gradient linear.
`TextureStretch`	Controls texture stretching along the curve. Value 1.0 means that the whole texture is repeated along the curve over the duration of one field.
`Flip`	Set to 1 to flip the opacity/thickness gradient. This could help to visualize the situation when the object flies towards the camera.

Power property

The power properties controls the shape of the gradient. See the following graph. The X axis is the position along the curve from the beginning to the end. The Y axis is the value that is used to module the parameter (either opacity or thickness). At the beginning of the trajectory, the parameter has full value that decreases up to the specified end value. The way the value decreases is determined by the power property.



Example

```
<Tool type="aki.paint.tools.trajectory.TrajectoryTool">
  <Id value="Trajectory"/>
  <Thumbnail value="tTrajectory.png"/>
  <ChromaKeyEnabled value="0"/>
  <Thickness value="2.5"/>
  <Texture value="line_dash.png"/>
  <PlaceType value="aki.productplacement.place.screen.PlaceModelScreenMotion"/>
  <OpacityStartLength value="3"/>
  <OpacityEndValue value="0"/>
  <OpacityEndPower value="2"/>
  <ThicknessEndFactor value="0.5"/>
  <ThicknessEndPower value="1"/>
  <TextureStretch value="0.1"/>
</Tool>
```

Tag

`aki.paint.tools.tag.TagTool`

Image or .gtc clip that is put into image by clicking a mouse. Can be dragged with a mouse after it is placed. Supports keyframe animation. Supports pausing at specified gtc frame, playing out the rest of the clip as a cut-out effect.

Tag graphics can display additional textual layers that's content that can change in time according to user-defined timeline.

`File`	Name of the tag resource file (image/clip).
`Size`	Float value defining size of the tag graphics. Value of 1.0 means that the graphics pixels have the same height as the background video pixels.
`ContentPath`	Path to a tag content resources
`AnchorPoint`	Reference point of the tag graphics. Values are in the <0,1> range. [0.5,0.5] by default.
`TextValue`	Text overlay static text content.
`ThumbnailTextYPos`	Y-position of the thumbnail text position, in pixels
`ThumbnailTextSize`	Size of the thumbnail text
`ThumbnailTextColor`	Color of the thumbnail text in rgba floats notation. Black by default.
`ThumbnailText`	Text rendered into the thumbnail
`RotationAngleDeg`	Rotation of tag graphics, in degrees.
`3DScale`	By enabling the value, the graphics are placed on 3D pitch, when available. Scale value defines default size of the graphics in 3D space.
`BillboardEnabled`	When enabled, the tag is rendered in a stand-up position when placed on pitch.
`BillboardOffset`	Offset from the ground up in meters.
`BillboardType`	`STATIC` - Camera position is determined at the drawing creation and then remains static (drawing is facing in the same direction). `DYNAMIC` - Camera position is determined dynamically throughout the drawing existence.

Font parameters

`TextPosition`	Offset of the text layer line relative to reference point, in background graphics pixels.
`TextPositionReference`	Reference point for the text position offset. Can be TOP_LEFT(default) or TOP_RIGHT of the background graphics.

TextSize	Height of one text line in pixels of the background graphics.
FontHAlign	Horizontal alignment { LEFT | CENTER | RIGHT | DOT_FIRST | DOT_LAST }
FontHAlignChar	Horizontal alignment character
FontVAlign	Vertical alignment { TOP | CENTER | BOTTOM | FIRST | LAST }
FontLineSpacing	Relative line spacing. Value 1.0 means spacing by the height of the font.
FontLineFlowPolicy	Behavior for lines wider then defined limit. { DONT_CARE | SCALE | UNIFORM_SCALE | SOFT_NEWLINE | HARD_NEWLINE | SOFT_SKIP | HARD_SKIP | SINGLE_LINE }
FontLineWidth	Line width limit. (see FontLineFlowPolicy)
FontFixedPitch	Fixed character pitch. (disables kerning defined by font).
FontKerning	Additional kerning.
FontCapitalsRatio	If enabled (value > 0) then all characters are made capitals. The original non capital letters are rendered smaller. This value defines a factor for that scale.
FontName	Name of the font. All system fonts are available.
FontResolution	Font texture resolution in pixels. It should approximately match the pixel size on the screen.
FontBaseColorMode	Coloring text mode. { SOLID | CHAR_4POINTS | LINE_4POINTS | GLOBAL_4POINTS | LINE_2GRADIENTS }
FontBaseColor	Color to be used when mode is SOLID
FontBaseColorP1	Color for gradient coloring
FontBaseColorP2	Color for gradient coloring
FontBaseColorP3	Color for gradient coloring
FontBaseColorP4	Color for gradient coloring
FontOutlineSize	Size of outline in pixels. Set to zero to disable outline.
FontOutlineColor	Outline color.
FontShadowOffset	Relative offset of shadow to normal text.
FontShadowBlur	Amount of blur of the shadow.
FontShadowColor	Shadow color.

Tracking: Screen, ScreenMotion, 3D
Keyframing: yes
Chroma key: no

Example

```
<Tool type="aki.paint.tools.tag.TagTool">
```

```
      <File value="Tags/tag.gtc"/>
      <PauseFrame value="15"/>
      <ThumbnailTextColor value="[.1, .1, .1, 1]"/>
      <ThumbnailTextYPos value="35" />
      <ThumbnailTextSize value="15"/>
      <ThumbnailText value="Tag"/>
      <ContentPath value="Tags"/>
      <AnchorPoint value="[0.5,0.4]"/>
</Tool>
```

Tackle Box

`aki.paint.tools.tacklebox.TackleBoxTool`

Textured rectangle drawn onto the pitch to highlight an area at the pitch. Displays color outline at the head side of the box.

Insertion is optimized to quickly set up the correct pitch perspective. Use following workflow to define the rectangle shape:

- A head line is defined by clicking and dragging the left mouse button over the video. A tackle box graphics appears in its default perspective.
- A manipulator shows up over the head line, following the mouse. Dragging the manipulator lets user define the first horizontal perspective line.
- Another manipulator following the mouse shows up to define the second horizontal perspective line.

Properties allow to define textured fill of the inner area and textured outline. Both textures can be colored using Color or ColorMatrix coloring. Setting empty texture filename results in solid color fill/outline.

Tracking: Motion
Keyframing: no
Chroma key: yes

Example

```
<Tool type="aki.paint.tools.tacklebox.TackleBoxTool">
      <TextureFile value="texture.png"/>
      <TextureLoopStart value="0"/>
      <OutlineThickness value="2"/>
      <OutlineTexture value="stroke.png"/>
      <OutlineColor value="[1, 1, 0, 1]"/>
      <FillColor value="[1, 1, 1, 1]"/>
</Tool>
```

Thumbnail coloring

Thumbnail coloring is controlled by ThumbnailColoringEnabled property. When is enabled, thumbnail pixels are colored with tool colors according to following table:

RED	FillColor
GREEN	Not used. Always white.
BLUE	OutlineColor

ZoomSpot

`aki.paint.tools.zoomspot.ZoomSpotTool`

Shows a spot shape (either circular or rectangular) with optional zoom effect inside. Saturation of video outside the spot can be dimmed.

`ColorMatrix`	Color mapping of the outer texture.
`Opacity`	Outer texture effect opacity.
`Size`	Size of the spot shape. In normalized screen coordinates.
`Zoom`	Magnification factor applied to the inward area of the spot shape.
`BWFactor`	Amount of de-saturation effect of the outward area. <0,1>.
`Shape`	CIRCLE or RECTANGLE. CIRCLE is default.
`RectangleAspect`	Aspect of the rectangle shape; ignored for CIRCLE.
`RimThickness`	Thickness of the spot rim, in normalized screen coordinates.
`Texture`	Texture image applied radially to the spot shape.
`TextureCoordinateSpace`	Texture image dimensions
`TextureSpotEdgePos`	X coordinate in texture for the inner edge. In texture image pixels.

Tracking: Screen, ScreenMotion
Keyframing: yes
Chroma key: no

Example

```
<Tool type="aki.paint.tools.zoomspot.ZoomSpotTool">
   <Id value="Zoom"/>
   <Zoom value="2.0"/>
   <Size value="0.35"/>
   <BWFactor value="0.5"/>
   <Thumbnail value="tZoom.png"/>
   <Texture value="zoomspot_line.png"/>
   <TextureCoordinateSpace value="[65,8]"/>
   <TextureSpotEdgePos value="7"/>
   <RimThickness value="0.018"/>
</Tool>
```

ZoomClip

`aki.paint.tools.zoomspot.ZoomSpotTool`

Shows a clip with optional zoom effect inside of the masked part of clip.

`File`	Path to gtc clip.
`Size`	Size of the clip. In normalized screen coordinates.
`Zoom`	Magnification factor applied to the masked area of the clip.
`Opacity`	Opacity multiplication constant.
`ColorMatrix`	Advanced clip coloring. Changed from GUI.
`ThumbnailOverlay`	Overlay part of thumbnail.

Tracking: Screen
Keyframing: yes
Chroma key: no

Example

```
<Tool type="aki.paint.tools.zoomclip.ZoomClipTool">
   <Id value="Zoom"/>
   <Zoom value="2.0"/>
   <Size value="0.35"/>
</Tool>
```

Zoom Pan

`aki.paint.tools.zoompan.ZoomPanTool`

Zoom Pan tool allows to select zoomed picture area in the video footage. Available both for Live and recorded footage.
Zoom Pan can be animated, stored into the clip, and replayed together with the clip.

There is a playback speed timeline integrated into the zoom pan tool. Timeline allows to schedule playback speed changes into the ZoomPan production. Additionally, the timeline shows ZoomPan keyframes for better overview of created animation.

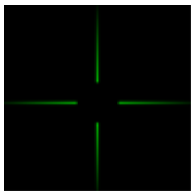
Pan
When the video is zoomed-in, panning is used to change the viewport of the video sequence. Simply drag the Video preview canvas with the left mouse button down.

Controls

Change the current zoom. Drag the thumb on the zoom scale.

Delete the current keyframe. Available only when a keyframe has been defined at the current timecode.

Zoom pivot. Click into preview to define the pivot. When not defined (not visible), the center of the screen is used as pivot.



Center of the screen.
The cross-hair may help to keep the tracked target in the center of the screen when creating an animation.

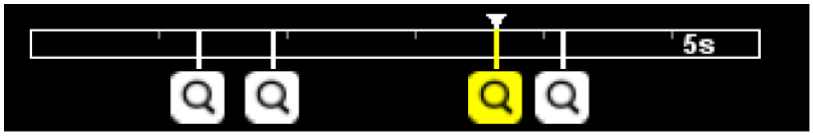
Save animation modifications.

Edit animation.

Animation Framework

- Define several keyframes with custom Zoom and Pan values. Transitions between different Zoom/Pan values are automatically interpolated.
- If there's an Zoom Pan animation in the clip, you have to press Edit animation button first. Clips don't support multiple animations but you can edit (extend) the existing one.
- Save the animation to the clip.

Zoom Pan timeline



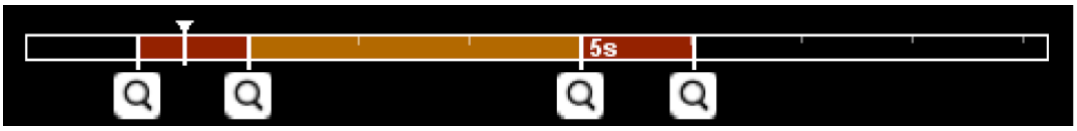
Zoom Pan Keyframe.
Click the keyframe to jump to the time of the keyframe.

Active keyframe.
Current time equals the time of the keyframe.

Time marker (current time).

Zoom Pan animations with Pause and Slow Motion blocks

You can use the Zoom Pan tool with Pause/Slow motion at once to create more advanced animations.
Example (Pause during Zoom-in and Zoom-out, Slow motion playback in-between).



More info is in chapter Playback speed timeline.

Supported features
Tracking: None
Keyframing: yes
Chroma key: no

Example of configuration

```
<Tool type="aki.paint.tools.zoompan.ZoomPanTool">
    <Thumbnail value="tZoomPan.png" />
</Tool>
```

Playback Speed

`aki.paint.tools.playbackspeed.PlaybackSpeedTool`

This tool allows to insert pause or slow motion blocks into clip playback.
More info is in chapter Playback speed timeline.

Example

```
<Tool type="aki.paint.tools.playbackspeed.PlaybackSpeedTool">
  <Thumbnail value="tPlaybackSpeed.png"/>
</Tool>
```

Blocking Line

`aki.paint.tools.freehand.PolyLineTool`

Special kind of Poly Line with blocking marker at the end. See Poly Line for list of properties.

Subtitles

`chyronhego.subtitlestool.SubtitlesTool`

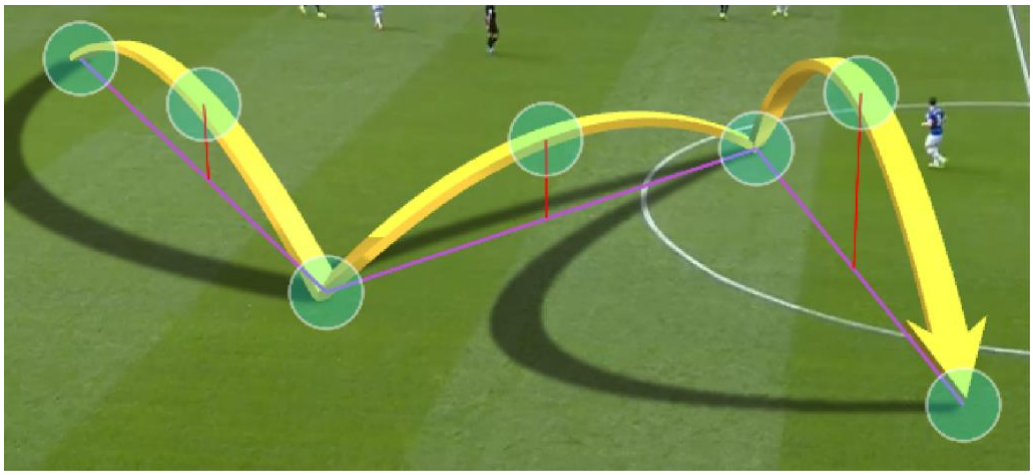
This tool allows you to create subtitles in the video area and save them into a clip.
To create a subtitle, just select the Subtitles tool to show the input/navigation dialog. The dialog is shown only when a clip is selected.
Type in the text of the subtitle and press Enter key. Duration of one subtitle is from the point of definition to the point of redefinition to a new value.

Tracking: Screen
Keyframing: no
Chroma key: no

Arrow Tool

`chyronhego.paint.tools.arrowtool.ArrowTool`

Arrow tool allows to create 3D arrows. First you have to set perspective for the clip – see 3D Calibration. Since 3D Pitch might be defined in clip only, this tool works in clip only as well.



The arrow is composed of segments called bounces. Each bounce is controlled by three points, two of which are at the ends of the bounce and lie on the ground. The third control point is between them and controls the height of the arc.
If **MultipleBounces** property is checked, each click into the preview appends a bounce into currently edited arrow.
In case a simple start-to-end arrow without any bounces is desired, uncheck the **MultipleBounces** property. In such case new arrows are created instead of appending bounces to the last one.

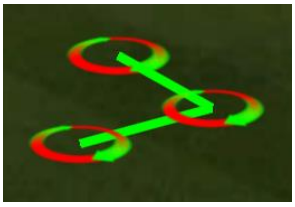
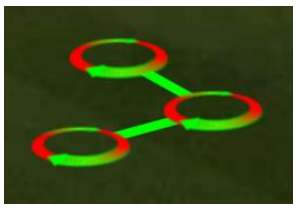
It is possible to display a text showing a distance between arrow end points. Distance text appearance is controlled from the Distance property tab. An important property is a **TrackingType** that decides how the distance text is presented:

NONE	On-screen distance text. Overall length of the arrow is shown. TextPosition defines the absolute position of the text, in normalized screen coordinates.
SHAPE_ALIGNED	Distance text in pitch plane. Shown for each arrow segment. TextPosition defines metric offset of text from the center point of arrow segment.
CAMERA_FACING	3D distance text standing on pitch. Shown for each arrow segment. TextPosition defines metric offset of text from the center point of arrow segment.

Distance text supports background texture. Background texture is controlled from the Distance Texture properties tab.

Tracking: 3D
Keyframing: yes
Chroma key: Off or Shadow only

Masked .gtc clip

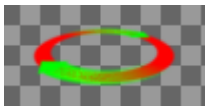
Tools combining cursors with stroke lines (LinkedCursors, CursorWithTrail) often utilize masked clip functionality. Masked clip prevents connecting stroke from showing through the cursor graphics.
Masked clip is a .gtc clip containing additional aux alpha channel that specifies where the underlying graphics should be suppressed.

To create a clip containing the mask channel, specify a mask image (image sequence) as a *Warp File Name* at the Clip Convertor Source File page. Additionally, check the *Subtract Aux Alpha* checkbox at the Output Video Parameters page.

Following mask file types are supported:

RGBA, IA	Image alpha channel is used as a mask.
Grayscale	Pixel intensity is used as a mask value (white – fully masked, black – no mask).
RGB	Image is converted to grayscale, pixel intensity is used as a mask.

Example: to create masked clip shown above, following inputs were used:

RGBA image: 

Mask grayscale image: 

Licenses

FreeImage Public License - Version 1.0

1. Definitions.
1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.
1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.
1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.
1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.
1.5. "Executable" means Covered Code in any form other than Source Code.
1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.
1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.
1.8. "License" means this document.
1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:
A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.
B. Any new file that contains any part of the Original Code or previous Modifications.
1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.
1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control
compilation and installation of an Executable, or a list of source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.
1.12. "You" means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or otherwise, or (b) ownership of fifty percent (50%) or more of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.
2.1. The Initial Developer Grant.
The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:
(a) to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, or as part of a Larger Work; and
(b) under patents now or hereafter owned or controlled by Initial Developer, to make, have made, use and sell ("Utilize") the Original Code (or portions thereof), but solely to the extent that any such patent is reasonably necessary to enable You to Utilize the Original Code (or portions thereof) and not to any greater extent that may be necessary to Utilize further Modifications or combinations.

2.2. Contributor Grant.
Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:
(a) to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code or as part of a Larger Work; and
(b) under patents now or hereafter owned or controlled by Contributor, to Utilize the Contributor Version (or portions thereof), but solely to the extent that any such patent is reasonably necessary to enable You to Utilize the Contributor Version (or portions thereof), and not to any greater extent that
may be necessary to Utilize further Modifications or combinations.

3. Distribution Obligations.
3.1. Application of License.
The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.
3.2. Availability of Source Code.
Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.
3.3. Description of Modifications.
You must cause all Covered Code to which you contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.
3.4. Intellectual Property Matters
(a) Third Party Claims.
If You have knowledge that a party claims an intellectual property right in particular functionality or code (or its utilization under this License), you must include a text file with the source code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If you obtain such knowledge after You make Your Modification available as described in Section 3.2, You shall promptly modify the LEGAL file in all copies You make
available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.
(b) Contributor APIs.
If Your Modification is an application programming interface and You own or control patents which are reasonably necessary to implement that API, you must also include this information in the LEGAL file.
3.5. Required Notices.
You must duplicate the notice in Exhibit A in each file of the Source Code, and this License in any documentation for the Source Code, where You describe recipients' rights relating to Covered Code. If You created one or more Modification(s), You may add your name as a Contributor to the notice described in Exhibit A. If it is not possible to put such notice in a particular Source Code file due to its
structure, then you must include such notice in a location (such as a relevant directory file) where a user would be likely to look for such a notice. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of
warranty, support, indemnity or liability terms You offer.
3.6. Distribution of Executable Versions.
You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.
3.7. Larger Works.
You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A, and to related Covered Code.

6. Versions of the License.
6.1. New Versions.
Floris van den Berg may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.
6.2. Effect of New Versions.
Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Floris van den Berg
No one other than Floris van den Berg has the right to modify the terms applicable to Covered Code created under this License.
6.3. Derivative Works.
If you create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), you must (a) rename Your license so that the phrases "FreeImage", `FreeImage Public License", "FIPL", or any confusingly similar phrase do not appear anywhere in your license and (b) otherwise make it clear that your version of the license contains terms which differ from the FreeImage Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.
COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.
This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

9. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE
EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THAT EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.
The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

11. MISCELLANEOUS.
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by Dutch law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in, the The Netherlands: (a) unless otherwise agreed in writing, all disputes relating to this License (excepting any dispute relating to intellectual property rights) shall be subject to final and binding arbitration, with the losing party paying all costs of arbitration; (b) any arbitration relating to this Agreement shall be held in Almelo, The Netherlands; and (c) any litigation relating to this Agreement shall be subject to the jurisdiction of the court of Almelo, The Netherlands with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys fees and expenses. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.
Except in cases where another Contributor has failed to comply with Section 3.4, You are responsible for damages arising, directly or indirectly, out of Your utilization of rights under this License, based
on the number of copies of Covered Code you made available, the revenues you received from utilizing such rights, and other relevant factors. You agree to work with affected parties to distribute responsibility on an equitable basis.

EXHIBIT A.
"The contents of this file are subject to the FreeImage Public License Version 1.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://home.wxs.nl/~flvdberg/freeimage-license.txt
Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

FFMPEG

This software uses code of FFmpeg licensed under the LGPLv2.1 and its source can be downloaded from the Download Area.