

Rio 4K/8K

High performance online editing,
color and finishing

Release Notes

Software Version v4.5.9 (October 2021)

Trademarks and logos

Grass Valley®, GV® and the Grass Valley logo and/or any of the Grass Valley products listed in this document are trademarks or registered trademarks of GVBB Holdings SARL, Grass Valley USA, LLC, or one of its affiliates or subsidiaries. All other intellectual property rights are owned by GVBB Holdings SARL, Grass Valley USA, LLC, or one of its affiliates or subsidiaries.

All other trademarks and logos used in this document are either registered trademarks or trademarks of the manufacturers or vendors of the associated products.

- Adobe, Adobe logo, Adobe Reader, Adobe PDF and Adobe PDF logo, are trademarks of Adobe Systems Incorporated.
- Apple ProRes is trademark of Apple Inc., registered in the U.S. and other countries.
- QuickTime and the QuickTime logo are trademarks or registered trademarks of Apple, Inc.
- Avid, Avid DNxHD, and Avid DNxHR are trademarks or registered trademarks of Avid Technology, Inc. or its subsidiaries in the United States.
- EOS is a trademark of Canon Inc.
- Dolby, Dolby Vision, and the double-D symbol are registered trademarks of Dolby Laboratories Licensing Corporation.
- Microsoft, Windows, and Internet Explorer are trademarks or registered trademarks of Microsoft Corporation in the United States and other countries.
- AVCHD is a trademark of Panasonic Corporation and Sony Corporation.
- P2, AVCCAM, AVC-Intra, and AVCUltra are trademarks of Panasonic Corporation.
- The trademark, service mark, product name logo, and product name of RED are trademarks or registered trademarks of Red.com, Inc. in the United States and other countries.
- XDCAM, XDCAM EX, XAVC, XAVC S and SxS are trademarks of Sony Corporation.

Grass Valley® Product Support

For technical assistance, to check on the status of a question, or to report a new issue, contact Grass Valley Product Support via email.

Web Technical Support

To access support information on the web, visit the product support webpage on the Grass Valley website. You can download software or find solutions to problems by searching our Frequently Asked Questions (FAQ) database.

- World Wide Web: <http://www.grassvalley.com/support/>
- Worldwide Technical Support: [Please use our online form](#)

Software Release Details

Release Description

V4.5 is a feature release for Rio 8K, 4K & 2K, Rio Assist and Rio Connect only. Some of the following features do not apply to all products

Release Type

General release that includes updates for Dolby Vision® V4 and internal CMU support, software scopes and preview with metadata overlay, with bug fixes

This version

V 4.5.9

Release Date

Oct 12, 2021

Other Notes

System Drivers – Please note that there are NO updates to drivers for this version over V4.3.0 except graphics driver. If upgrading from earlier versions, Please see page 4 for more information.

Base, Desk and Dict folders are compatible with V4.3.0 and V3.1 rev 1. This database schema is NOT compatible with V3.0 rev 1 or earlier.

Product Overview

The Rio range offers a choice of software or turnkey high quality editing, color and finishing systems running on standard PC hardware. The Rio range provides SD, HD, 2K, 4K, 6K, 8K stereo3D and high-frame rate support.

The comprehensive integrated toolset includes conform, color, editing, effects, text, paint and multi-format versioning enabling Rios to complete a wide range of post production tasks. Rio efficiently handles today's digital capture formats such as RED, XAVC, Sony F65, SStP, Canon, Apple ProRes and easily integrates into data-centric post pipelines.

Installation

Windows 10 64bit is required to run this software. An engineer may be required to do this upgrade.

You will need to run the installation program, setup.exe, and the software as administrator.

Required Software

Name	Description
DirectX	x64 application requires DirectX 11 to be installed... the installer will install if the system is found to be running an earlier version of DirectX
Internet Explorer	Internet Explorer 11 (even if not network connected) needs to be installed or else the software will not run up
DotNet Framework	Microsoft .NET Framework 3.5 or later
Adobe Acrobat Reader	Acrobat Reader DC 20.009.20067

Associated or optional software

Name	Description	Version
Dolby Vision CMU	Dolby Vision external unit	2.6 or later
Neo Workstations Server	GV software for Neo/Nano Workstation PC	V4.5.9
Media Trasformer	'deliver to' connectivity for sQ server systems	V5.4 rev 5
edlaaf.exe	CMX 3600 list converter to AAF	V54.0
aaf2edl.exe	AAF list converter to CMX 3600 format	V2.14
nclaaf.exe	Negative Cut List converter to AAF	V10.0
stlaaf.exe	Subtitle List Converter to AAF	V3.0
XMLDir.exe	QXMLNet clip bin query utility	X5.4.0
XMLTool.exe	QXMLNet clip combining utility	X5.4.0
XMLPut.exe	QXMLNet clip import utility	X5.4.0
XMLGet.exe	QXMLNet clip export utility	X5.4.0
XMLArchIn.exe	QXMLNet archive in utility	X5.4.0
XMLArchOut.exe	QXMLNet archive out utility	X5.4.0

Driver Requirements

Name	Version
Sentinel Driver	V7.6
nVidia Tesla driver (K, P series)	442.50-tesla-desktop-win10-64bit-international
nVidia GeForce GTX driver	442.50-desktop-win10-64bit-international-whql
nVidia Quadro driver (Quadro and M series)	442.50-quadro-desktop-notebook-win10-64bit-international-whql
AJA Corvid Ultra driver	ntv4driver-2.5.0.253.msi , Firmware v65
AJA Kona 3G driver	ntv2driver-12.3.7.85.msi , Firmware v124
AJA Kona 4 driver	ntv2driver-12.3.7.85.msi , Firmware v44
AJA Corvid 88 driver (4K and 8K Rio)	ntv2driver-12.1.4.146.msi , Firmware v25 (from v3.1 rev 1 installer)
eVidIO1 card driver	v17
PCi QLink driver	3.3.0.0
Wacom tablet driver	WacomTablet_6.3.9w3.exe – Windows Driver 6.3.39-1
Serial Card (VTR) driver	CDM 2.06.00 WHQL Certified.zip
RED Rocket-One driver	RED driver 2.1.23.0, Firmware 1.1.18.0
RED Rocket-X driver	RED driver 2.1.23.0, Firmware 1.4.1.16

IMPORTANT: On boot of the Rio software the user will be prompted to update the AJA output device driver and firmware if it has not already been done.

Driver and matched firmware for each software build can be found in...

C:\Data\{Product}\DLL_Data\Quantel\AVIO3PCardDriver

Install instructions for Driver and Firmware can be found in the above folder (readme.txt)...

NOTE: Corvid 88 driver ntv2driver – 12.3.7.85 and Firmware v32 has an issue with horizontal line noise – GENQ-22773

Therefore we advise that all Corvid 88 installations are DOWNGRADED to driver ntv2driver-12.1.4.146 and Firmware v25.

The above 'downgrade' version is not included in V4.5.9 software.

Driver and firmware must be installed from V3.1 or 3.0 software.

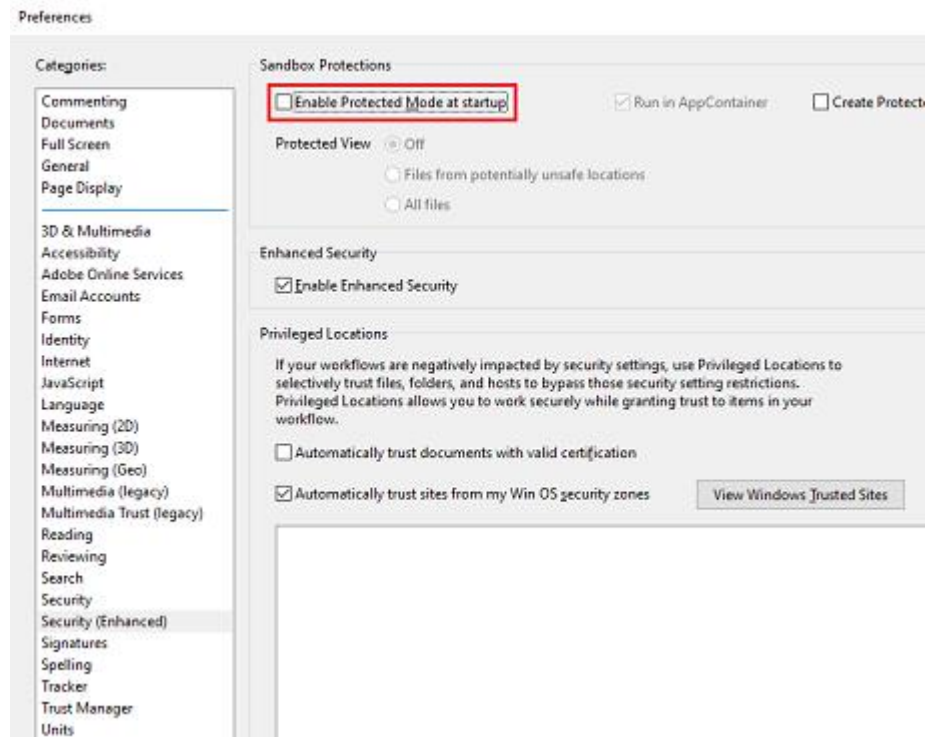
(this issue ONLY affects AJA Corvid 88 hardware)

Change Acrobat Reader DC option

Once the Rio upgraded including required driver / software upgrade, follow the below steps to change Acrobat Reader DC option.

- 1) Run Acrobat Reader DC
- 2) Select Edit > Preferences
- 3) Select "Security (Enhanced)" of Categories

- 4) Uncheck "Enable Protected Mode At Startup."



- 5) Click [OK] button
- 6) Exit Acrobat Reader DC

Enable option for Wacom Pen Display

If you use Wacom Ciintiq Pro series Pen Display, run the Rio settings then enable required option:

- 1) Quit Rio software if running
- 2) Choose Start Grass Valley > Quantel Rio V4.5.9 Settings
- 3) Tick 'Enable embedded LCD tablet mode' option



- 4) Click [Apply] then [OK] button
- 5) Start Rio software

Changes this Release

New Features and Enhancements in V4.5.9

- Waveform / Vectorscope support in Rio
- Support importing 12-bit DPX file generated with Canon Cinema RAW Development software
- Dolby Vision upgraded to 5.1.1 including Internal CMU support and support for External CMU 5.1.0 firmware.
- Cascade Export - export all Cascades, Preprocess and CDLs as a single LUT
- Metadata (Timecode) Overlay on Preview
- Support exporting CDL transforms for the entire timeline (SFDC00759553)
- Add an option to control whether alpha is un-premultiplied on EXR import (SFDC00749451)
- Support for USB 3 connection to the Neo/Nano Workstation PC

Resolved issues

- Rio crashes if the timeline and a floating clip are played in sync just after the application is launched (SFDC00744664)
- RIO v4.5.8 cannot import a certain mov file (SFDC00752994)
- Incorrect color appears when importing with DNxHD 350 x codec (SFDC00753051)
- Rio incorrectly detects scan type when loading interlaced AAF (SFDC00737650)
- Rio corrupts / cuts Sony F65 MXF file (4096x2160) to be 3840x2160 while importing the file (SFDC00761021)
- RIO v4.5.8 handles respeeded and rendered clip as unrendered clip (SFDC00761177)
- Deinterlace fields are inverted for NTSC SD clips
- Old Effects behaviour to be consistent with regards to colour spaces when resizing
- 12bit DPX files have red - blue swap (SFDC00764747)
- Support exporting 12bit DPX files (SFDC00764748)
- Support to import 12bit DPX files generated by DaVinci Resolve (SFDC00764748)
- Archived YUV444 clips cannot be restored properly (SFDC00762439)
- Space Analysis Report was not created if the Logs folder was not on C: drive
- Rendering with Matinee works too slow in Rio V4.5.8 (SFDC00766946)
- Rio doesn't import 4K ProRes 422 HQ clips (SFDC00759738)
- MLT FX Radius Y key frame doesn't appear
- Archived SD clip's color is incorrect (SFDC00767592)
- QPlugin - pass the rush filepath through to plugins via auxiliary metadata
** This is for improvement of automation workflow*
- Float clips fail to archive to tape if EXR compression has been used
- Rio restripes clip TC if import source has odd start TC (SFDC00698298)

- Rio crashes while importing ProRes 444 mov file (SFDC00768532, 00770193)
- Rio is unable to import YUV/10bit when localizing (SFDC00777510)
- RIO v4.5.8 crashes when adding OFX effect to blank
- Another picture unexpectedly mixed in soft-mounted 8K clip (SFDC00782803)
- IO, Export Pro Res exports change audio track mapping from stereo to mono (SFDC00785979)
- Rio application working becomes to slow after conform (SFDC00785609)
- Rio is unable to import YUV/10bit when localizing DNxHR files (SFDC00788082)
- When importing an AVI file, imported clip has wrong frame rate and start timecode is missing (SFDC00795690)
- Rio crashes when adding 59.94p 32ch clips to timeline

Known issues and limitations

Known issues

- I/O Export, alpha channel image is burned to exported fill video by HQHGX
- "Module jpeg2000 - bpno became negative" error message appears in log when loading an mp4 JPEG2000 file

Design Limitation

Restrictions after Uninstallation of QuickTime for Windows

The following file import support will be lost if QuickTime is uninstalled:

- .aif, .aiff : Audio Interchange File Format
- .dv : DV video and audio files
- .gif : GIF graphic files
- .mp2, .mp3 : MPEG-1/2 audio

Supported New Features

1. Dolby Vision®

Overview

Rio now supports the Dolby Vision Content Mapping V4.0 algorithm, as well as the previous version V2.9. The V4.0 workflow is fully backwards compatible with V2.9 and is recommended for new projects. It is not possible to upgrade an existing project using V2.9 to use V4.0 as the timeline will need to be re-analysed, erasing existing trim data.

Rio no longer requires clips to be rendered before Dolby Vision analysis and trim passes. Clips can stay unrendered and be rendered on export, either the master with Dolby Vision XML mastering metadata or as a target trim using the Internal CMU feature.

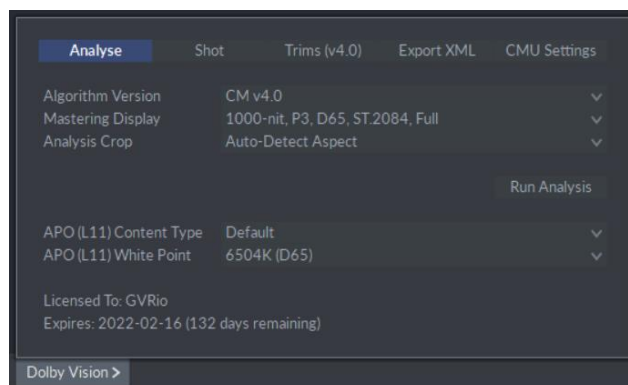
Rio supports both current External CMU protocols – the original v3 CMU firmware only supports V2.9 projects, but the new v5 firmware supports both V2.9 and V4.0 projects. Alternatively, the new Internal CMU feature applies the Dolby Vision content mapping inside Rio and does not require an External CMU device.

License & Configuration

A license from Dolby is required to adjust individual trim controls. Without the license you can perform the analysis process but the Trims tab will not appear. Licenses can be obtained from dolbyvisionmastering@dolby.com. The license file should be placed in **C:\Data\Licenses** ensuring that the file has the “.bin” extension.

Dolby may also produce custom configuration files to add support for new Mastering or Target displays. These are also provided in the form of “.bin” files which should also be added to the **C:\Data\Licenses** folder.

Analysis



To start using Dolby Vision the timeline first needs to be analysed. From the Edit menu click the Dolby Vision button to bring up the Analyse tab where you can configure the settings and then click Run Analysis to start processing the timeline. The following settings are available:

- Algorithm Version – choose either Content Mapping V2.9 or V4.0 workflow
- Master Display – set the HDR mastering display’s colour space and peak luminance

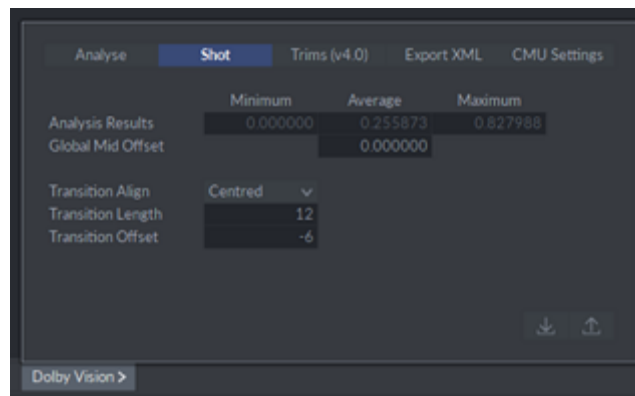
- Analysis Crop

- None – use the entire frame for analysis
- Safe Area – use the safe area set in the F3 menu for analysis
- Auto-Detect Aspect – automatically detects the aspect ratio using the current frame
- Auto-Detect Each Segment – automatically detects the aspect ratio for each segment
- Set Aspect Manually – manually set the aspect for analysis

To analyse only a portion of the timeline set in- and out-points before starting the analysis. The range may be expanded to ensure that whole segments are analysed.

This tab also allows you to set the Level 11 metadata for devices that support Automatic Picture Optimisation. This metadata consists of a Content Type and the Intended White Point of the mastering display.

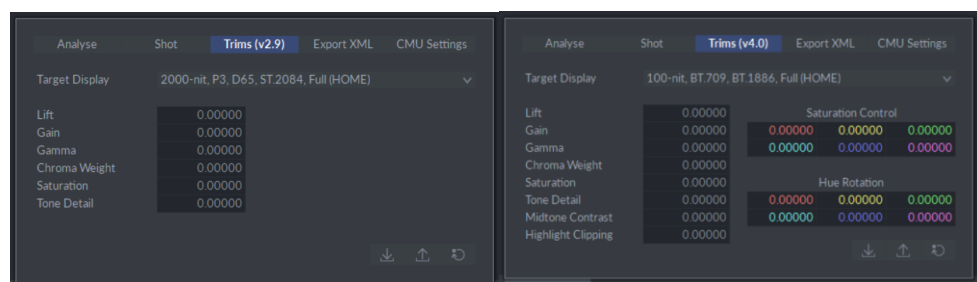
Per-Shot Metadata



The Shot tab shows you the results of the analysis pass, and for V4.0 workflows allows you to change the Global Mid Offset for all trims in this shot. The Store and Load buttons allow you to copy the L1 and L3 analysis results to a different segment.

During the analysis process transitions will be automatically created on the Dolby Vision data track to match video transitions on the timeline. The Shot tab allows you to resize these transitions, remove or create new transitions between shots.

Trim Controls



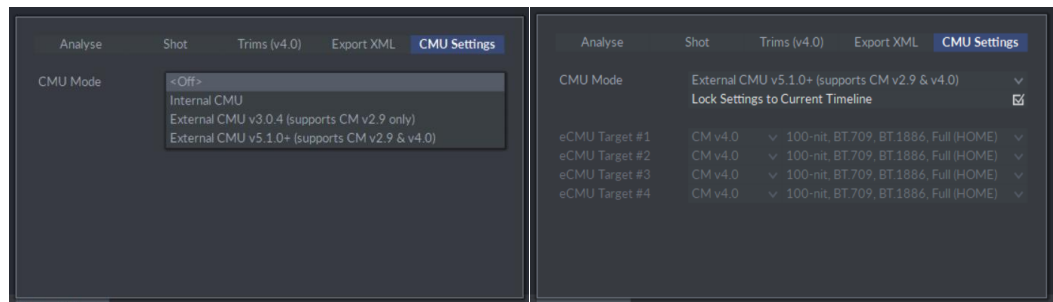
Once a clip has been analysed, the Trims tab allows you to tweak trim controls for each shot to adjust the look on a given target display. You do not need to make adjustments to all target displays as the trim controls are automatically interpolated for displays that have

not been explicitly set.

When using the V4.0 workflow there are additional parameters available compared to V2.9 that allow you to adjust hue and saturation more accurately using six colour channels (red, yellow, green, cyan, blue, and magenta).

The Store and Load buttons allow you to copy trim settings between segments; the Reset button resets all parameters to their default values. When using the Neo panel you can Store/Load up to 6 presets.

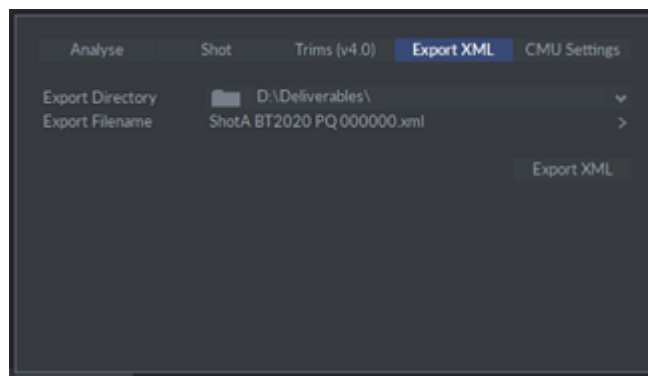
CMU settings



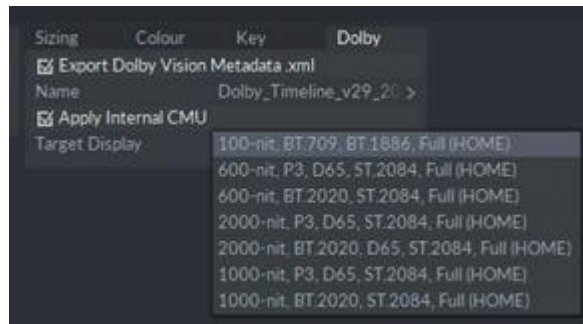
The CMU Settings tab allows you to control the Internal CMU and External CMU workflows. In the Internal CMU mode Rio applies the Content Mapping and all outputs and the screen are converted to the specified target display. In the External CMU mode Rio outputs the HDR source with Dolby Vision metadata on the SDI outputs so that External CMU devices can apply the content mapping.

You can use the “Lock Settings to Current Timeline” checkbox to match CMU settings from the current shot and current trim that you are editing. Alternatively you can manually set each CMU’s settings to view different target displays or to check how the clip looks using the backwards-compatible V2.9 mapping. Up to 4 External CMUs are supported and each can be set to a different target display.

Export XML / Apply Internal CMU



Once the Trim passes have been completed you can export the Dolby Vision mastering metadata to an .xml file for delivery. This can be done either in the Export XML tab from the Edit menu, or as part of the File Export process.

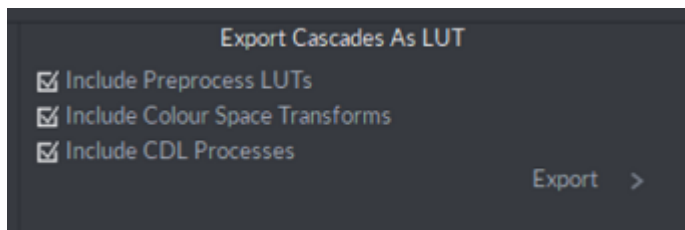


In the File Export menu there is an additional “Dolby” tab for clips that have Dolby Vision tracks. These settings allow you to export the metadata .xml file alongside the deliverable, or alternatively to use the Internal CMU process to export a deliverable for the specified target display.

The Internal CMU process does not require that the clip is rendered and will apply the render-on-export process before doing applying the content mapping.

2. Cascade Timeline Export

In the MLT Colour menu there is a new addition to the Preprocess tab. Previously you were only able to export the Base cascade as an external LUT file for use elsewhere in Rio or in third-party software. This functionality has now been extended so that you can export the complete colour pipeline of a given frame to a single LUT file.



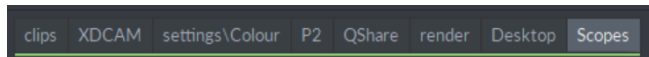
By default this exports all cascades, LUTs and CDL effects applied to the current frame. The checkboxes allow you to control what is included:

- Include Preprocess LUTs – also exports any LUTs found in Preprocess or Cineon Utils as part of the cascade export
- Include Colour Space Transforms – when exporting multiple cascades and using either Rio’s managed or ACES pipelines, include the automatic colour space transforms as part of the LUT file
- Include CDL Processes – in addition to Colour processes, also export CDL processes to the LUT file

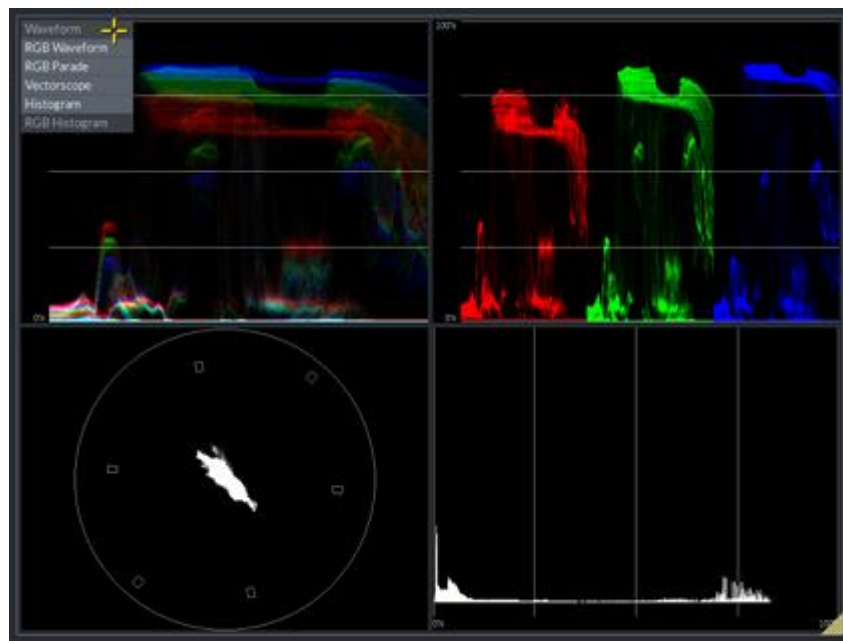
To export the file, click the Export button and type a filename. LUTs can be exported to different file formats depending on the extension given (for example .clf or .cube).

3. Scopes

Rio now includes built-in Scopes that can be found alongside the bins in the bottom launch bar.



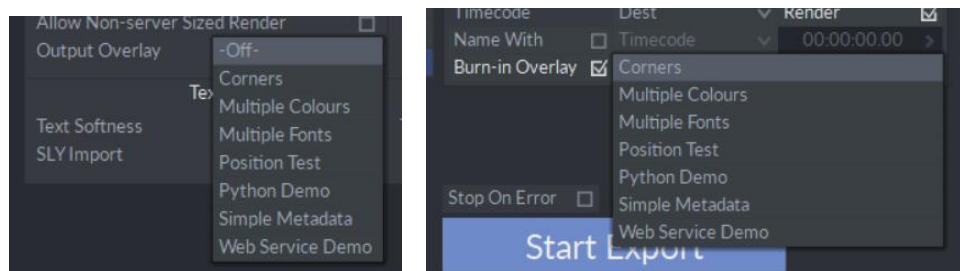
By hovering over the top left of the Scopes window you can control which scopes are visible. A maximum of four scopes can be enabled at once. The window can be resized by dragging its bottom-right corner, and will automatically switch between horizontal, vertical and square layouts.



4. Rio Metadata Overlay

Overview

Rio now supports text overlays, both on the SDI Output during playout and burnt-in during File Export. To enable the Output Overlay, open the F1 menu and choose a template from dropdown box; to burn in the overlay on export, choose a template from the File Export menu.



Overlay templates are read from the **C:\Data\User\OutputOverlays** folder, and a number of examples are placed there during the first install. You can place additional .txt text files in this folder and they will appear as templates the next time the software is started. (Text files containing Unicode characters must be saved in UTF-8 encoding.)

When a template is enabled, it is parsed for *commands* which control the position and formatting of the text on the output and for *metadata parameters* which are substituted for their values from the current clip, segment and frame.

In addition to plain text files, you can create .py Python files or .url Internet shortcut files to enable dynamic scripting support. These scripts and web services allow you to display a different template for each segment. For example, you could check an external database to see if a “Pending VFX” label needs to be displayed.

Metadata

Each line of text may contain metadata fields that are substituted into the overlay. For example, the line:

\$(clip.title) – frame \$(clip.frame) of \$(clip.duration)

may be displayed as:

Rio Test Clip – frame 303 of 10000

Here is a list of metadata parameters that can be used in this way:

Metadata Name	Description	Example
clip.duration	Duration of the timeline clip in frames	12345
clip.fps	Frame rate of the timeline clip	29.97
clip.tc	Destination Timecode (for first frame of clip)	01:20:30:11
clip.title	Title of the timeline clip	Clip Title
clip.category	Category of the timeline clip	Clip Category
clip.owner	Owner of the timeline clip	Clip Owner
clip.width	Frame width from the clip's output/render setting	1920
clip.height	Frame height from the clip's output/render setting	1080
seg.blank	Only present when the segment is blank	true
seg.start	Frame offset in clip (1-based) of start of segment	1234
seg.startz	Frame offset in clip (0-based) of start of segment	1233
seg.duration	Duration of the segment in frames	99
seg.tc	Source Timecode (for first frame of segment)	10:00:00:00
seg.title	Title of the current segment	Segment Title
seg.originator	Originator for the current segment	Originator
seg.filepath	File path associated with the current segment	D:\example.mp4
seg.fileoffset	Offset within file associated with first frame of segment	100
seg.width	Frame width of the segment's source rush	3840
seg.height	Frame height of the segment's source rush	2160

The following additional parameters are also available but are not passed to Python scripts or Web service calls. See those sections below for more information.

Metadata Name	Description	Example
clip.frame	Current frame number within clip (1-based)	12345
seg.frame	Current frame number within segment (1-based)	123
clip.framez	Current frame number within clip (0-based)	12344
seg.framez	Current frame number within segment (0-based)	122
clip.tc	Current frame's destination timecode	01:20:30:11
seg.tc	Current frame's source timecode	10:00:00:00

Commands

In addition to parameter substitutions, lines may begin with any number of commands. These commands control how the text line is displayed on screen. Commands take the form:

`$(command=argument)`

Commands affect their current line and all subsequent lines: for example, changing the font changes the current lines and all following lines until the font is changed again.

Here is a list of commands available:

Command	Description	Example
font.name	Sets the name of the font to use	<code>\$(font.name=Arial)</code>
font.size	Changes the font size	<code>\$(font.size=40)</code>
align	Changes the horizontal alignment (left/centre/right or l/c/r)	<code>\$(align=right)</code> <code>\$(align=c)</code>
x	Sets the horizontal (x) position	<code>\$(x=100)</code>
y	Sets the vertical (y) position	<code>\$(y=50%)</code>
line	Offsets the vertical position by N lines	<code>\$(line=-2)</code>
fg	Sets the text/foreground colour	<code>\$(fg=red)</code>
bg	Sets the background colour and transparency	<code>\$(bg=#44444480)</code>
expire	For Python scripts/Web service calls, sets the length of time to cache this segment's output. Set to zero to cache forever	<code>\$(expire=0)</code>

Text Position

Position can be set using either absolute pixel coordinates or relative percentage coordinates. Negative pixel values can be used to offset relative to the right/bottom borders. This example uses negative pixel values to display four pieces of metadata in the four corners of the frame:

```
$(x=100)$(y=100)$(align=l)$(clip.title)
$(x=-100)$(y=100)$(align=r)$(clip.tc)
$(x=100)$(y=-100)$(align=l)$(seg.filepath)
$(x=-100)$(y=-100)$(align=r)$(seg.tc)
```

The line command is not normally needed, as each line that does not contain an `$(y=)`

command is automatically offset relative to the previous by one line height. However it could be used to write lines upwards relative to the bottom of the frame, for example:

```
$(x=10%)$(y=90%)$(clip.tc)
$(lines=-1)$(clip.title)
```

Colours

The foreground and background colours can be specified in hex using either the #RRGGBB or #RRGGBBAA formats (where AA is an alpha channel). In addition, the following sixteen standard colours are available by name:

white	silver	gray	black
red	maroon	yellow	olive
lime	green	aqua	teal
blue	navy	fuchsia	purple

Python Script

By creating a .py Python script in the **C:\Data\User\OutputOverlays** folder you can create a template that behaves differently for each segment. Rio passes all of the segment metadata parameters to your script via the standard input ("sys.stdin" in Python) and then reads the template back from the script via standard output ("sys.stdout" or "print") unless it receives the command "\$ (end)". The template that is returned from the script may contain commands and metadata parameters for substitution exactly like the .txt format.

Here is a small example which requires a Python 3 interpreter to be installed on your computer. You can use this as the starting point for your own scripts by modifying the printOverlay() function.

```
import sys

def printOverlay(params):
    print("clip.frame = $(clip.frame)")
    for key, value in params.items():
        print(key, "= $" + key + ")")

def readParams():
    params = {}
    for line in sys.stdin:
        line = line.strip()
        if line != "":
            key, value = line.split("=", 1)
            params[key] = value
    else:
        return params
    sys.exit(0) # software was closed
```

```
while True:
    printOverlay(readParams())
    sys.stdout.write("$(end)")
    sys.stdout.flush()
```

When Rio first starts it attempts to locate a default Python installation on your machine. If your Python interpreter is version 3, it is important that all template scripts use Python 3 and not Python 2. If Rio cannot find an installation or if it chose the incorrect version, you can manually specify the Python location with the registry key

"Quantel\OutputOverlay\1.0.0.x\Custom Settings\Python Interpreter". It is recommended to use the "pythonw.exe" as this does not pop up a console window.

Web Service

If you create a .url Internet shortcut file in the **C:\Data\User\OutputOverlays** folder pointing to a web server, Rio will call your web service for each segment to request the overlay template. You can often create Internet shortcuts by dragging from your web browser's address bar to Windows Explorer, but alternatively you can manually create a file with the .url extension and the following text (changing the URL to match your web server):

```
[InternetShortcut]
URL=http://localhost:5000/
```

Rio will call the web service at the specified URL using an HTTP POST and providing the segment metadata as form data. The web service should respond with a "text/plain" document containing the template to render. This template can contain commands and metadata substitutions exactly like the .txt format. The URL may also contain metadata substitutions.

Custom Metadata

Using XmlPut to import files allows you to add Custom Metadata to clips. This custom metadata consists of key/value pairs along with a Customer ID (in GUID format) that groups your metadata in the database. If you have not used this feature before you can generate your own GUIDs.

```
XmlPut -cusid {3a2c2dd8-fa10-4607-bf3a-371522f8a61b}
        -auxkey camera -auxval B
        -auxkey final -auxval true
```

If you imported a clip using the above command then that segment on the timeline would have the following metadata parameters available for templates, Python scripts or Web services:

```
$( {3a2c2dd8-fa10-4607-bf3a-371522f8a61b }.camera)      B
$( {3a2c2dd8-fa10-4607-bf3a-371522f8a61b }.final)      true
```