Viz Mosart Web Applications

Version 3.5



Copyright ©2026 Vizrt. All rights reserved.

No part of this software, documentation or publication may be reproduced, transcribed, stored in a retrieval system, translated into any language, computer language, or transmitted in any form or by any means, electronically, mechanically, magnetically, optically, chemically, photocopied, manually, or otherwise, without prior written permission from Vizrt. Vizrt specifically retains title to all Vizrt software. This software is supplied under a license agreement and may only be installed, used or copied in accordance to that agreement.

Disclaimer

Vizrt provides this publication “as is” without warranty of any kind, either expressed or implied. This publication may contain technical inaccuracies or typographical errors. While every precaution has been taken in the preparation of this document to ensure that it contains accurate and up-to-date information, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained in this document.

Vizrt’s policy is one of continual development, so the content of this document is periodically subject to be modified without notice. These changes will be incorporated in new editions of the publication. Vizrt may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time. Vizrt may have patents or pending patent applications covering subject matters in this document. The furnishing of this document does not give you any license to these patents.

Antivirus Considerations

Vizrt advises customers to use an AV solution that allows for custom exclusions and granular performance tuning to prevent unnecessary interference with our products. If interference is encountered:

- **Real-Time Scanning:** Keep it enabled, but exclude any performance-sensitive operations involving Vizrt-specific folders, files, and processes. For example:
  - C:\Program Files\[Product Name]
  - C:\ProgramData\[Product Name]
  - Any custom directory where [Product Name] stores data, and any specific process related to [Product Name].
- **Risk Acknowledgment:** Excluding certain folders/processes may improve performance, but also create an attack vector.
- **Scan Scheduling:** Run full system scans during off-peak hours.
- **False Positives:** If behavior-based detection flags a false positive, mark that executable as a trusted application.

Technical Support

For technical support and the latest news of upgrades, documentation, and related products, visit the Vizrt web site at www.vizrt.com.

Created on

2026/11/19

Contents

1 Prerequisites 11
2 Getting Started 12
2.1 User Guides 13
3 Related Documents 14
4 Feedback and Suggestions 15
5 Configuration Tool 16
5.1 Opening the Configuration Tool 16
5.1.1 Who can open it, and who can change settings 16
5.1.2 The Configuration Tool window 17
5.2 Saving your changes 18
5.3 Home 19
5.4 Quick setup 19
5.5 Help & Resources 20
5.6 Troubleshooting 21
5.7 App Configuration 22
5.7.1 NRCS Plugin 22
5.7.2 Showmaker 24
5.7.3 Named Overlays Editor 29
5.7.4 Template Editor (Early Access) 29
5.7.5 Smart Panel 30
5.8 Studios 32
5.8.1 Defining studios 32
5.8.2 Studio support per application 33
5.8.3 Editing a studio's Mosart server settings 34
5.9 Server Configuration 35
5.9.1 Server addresses 35
5.9.2 Serve Config Tool on the web 35
5.9.3 Sign-in and sessions 36
5.9.4 Mosart API Access 36
5.9.5 Media Sequencer 37
5.9.6 Viz Arc 37
5.9.7 Pilot Data Server 37
5.9.8 Video Clips 38

5.9.9 TriCaster Live Previews 38
5.10 HTTPS and Certificates 39
5.10.1 Proxy setup method 39
5.10.2 Certificates 40
5.10.3 Mosart servers over HTTPS 42
5.11 Authentication 43
5.11.1 Prerequisites 43
5.11.2 Roles 43
5.11.3 Setup 45
5.11.4 Session timeout 46
5.11.5 Provider-specific newsroom authentication 47
6 NRCS Plugin 49
6.1 Working with the NRCS Plugin 50
6.1.1 User Operations 50
6.1.2 Keyboard Shortcuts 56
6.1.3 Getting Help 56
6.1.4 Error Handling 57
6.1.5 Session and Sign-in 57
6.1.6 API Documentation 58
6.2 NRCS Plugin in Pilot Edge 59
6.2.1 Hosting NRCS Plugin within Pilot Edge 59
6.3 Setup and Administration 61
6.3.1 Setting up the NRCS Plugin 61
6.3.2 Optional Server Setups 61
6.3.3 Session Timeout 62
6.3.4 Verifying the Mosart Installation 62
6.3.5 Template Type Filter Order 62
6.3.6 Clip Search 63
6.3.7 Hide Effects from Users 63
6.3.8 Integrating with an NRCS 64
6.3.9 Troubleshooting 67
7 Timing Display 68
7.1 Moving from Timing Display to Smart Panel 68
7.1.1 Rebuilding your displays 69
7.1.2 Note for Administrators 69
7.2 Documentation for Earlier Versions 70
8 Rundown Viewer 71

8.1 Working with the Rundown Viewer 72
8.1.1 Opening the Rundown Viewer 72
8.1.2 Key Features 72
8.1.3 Session and Sign-in 75
8.2 Setup and Administration of RV 77
8.2.1 Setting up the Rundown Viewer 77
8.2.2 Optional Server Setups 78
8.2.3 Troubleshooting 79
9 Showmaker 80
9.1 About Showmaker 80
9.1.1 Typical Usage 80
9.1.2 Key Features 80
9.2 Why Showmaker 81
9.2.1 Typical Workflow 81
9.3 Showmaker in your Ecosystem 81
9.4 Getting Started 83
9.4.1 Opening Showmaker 83
9.4.2 Tour of the Interface 84
9.4.3 Key Terms 85
9.4.4 Show Settings 86
9.4.5 Your First Rundown 87
9.4.6 Getting Help 88
9.5 Working with Showmaker 89
9.5.1 Common Actions 90
9.5.2 Landing Page 91
9.5.3 Show Assets 93
9.5.4 Rundowns View 93
9.5.5 Story Library 95
9.5.6 Rundown Operations 96
9.5.7 Timeline View 100
9.5.8 Story Properties 103
9.5.9 Story Approval 105
9.5.10 Show Settings 106
9.5.11 Collaborating on a Rundown 107
9.5.12 Timing from the Script 107
9.5.13 Recording a Story 108
9.5.14 Quick Guide to Rundown Operations 108

9.5.15 Rundown Templates View 109
9.5.16 Keyboard Shortcuts 109
9.5.17 Examples, Videos, Further Guidance 113
9.6 Setup and Administration of Showmaker 114
9.6.1 System Administrator Only 114
9.6.2 Setting up Showmaker for Daily Use 114
9.6.3 Optional Server Setups 121
9.6.4 Troubleshooting 121
10 Smart Panel 122
10.1 Why Smart Panel 122
10.2 Where Smart Panel Fits 122
10.2.1 Optional integrations 123
10.3 Key Features 123
10.4 Typical Workflow 124
10.5 Get Started 125
10.5.1 Open Smart Panel 125
10.5.2 Tour of the Interface 126
10.5.3 Key Terms 127
10.5.4 Your First Panel 129
10.6 Working with Smart Panel 130
10.6.1 Managing Panels 130
10.6.2 The Editor 131
10.6.3 Item Reference 135
10.6.4 Running a Panel On Air 148
10.6.5 Keyboard Shortcuts 150
10.7 Setup and Administration of Smart Panel 152
10.7.1 Setting up Smart Panel 152
10.7.2 Live Video Sources 154
10.7.3 Integrations for Button Actions 155
10.7.4 Roles and Access 155
10.7.5 Pairing Devices 156
10.7.6 Multiple Studios 157
10.7.7 Using HTTPS 157
10.7.8 Troubleshooting 157
11 Template Editor (Early Access) 161
11.1 Info: Early Access 161
11.2 Safety first 161

11.3 Template storage 161
11.4 Control access 161
11.5 Template Editor 161
11.6 Benefits of the new Template Editor 161
11.7 How the new Template Editor Works 162
11.8 Further Information 163
11.9 Working with Template Editor 164
11.9.1 The Dashboard 164
11.9.2 The Editor 165
11.9.3 Keyboard Shortcuts 167
11.10 Setup and Administration of TE 170
11.11 Command Reference 171
11.11.1 Video Switcher Commands 172
11.11.2 Audio Mixer Commands 176
11.11.3 Robotic Camera Commands 179
11.11.4 Light Commands 180
11.11.5 GPI/O Commands 181
11.11.6 Weather Commands 182
11.11.7 Graphics Commands 182
11.11.8 Generic REST Commands 183
11.11.9 Fine-tuning On-Air Action Timings 185
11.11.10 Template Terms and Definitions 186
11.12 Transitioning from the old Template Editor 187
11.12.1 Mapping Desktop Template Editor Commands to the Web Apps Template Editor 187
11.12.2 Currently Unsupported Command Properties 188
11.12.3 Examples 192
11.12.4 Legacy commands 195
12 Prompter App 199
12.1 Typical Usage 199
12.2 Prerequisites 199
12.3 Working with the Prompter 200
12.3.1 Opening the Prompter 200
12.3.2 The Prompter View 200
12.3.3 Following the On-Air Rundown 202
12.3.4 The Control Bar 203
12.3.5 Settings 203
12.3.6 The Reading Cue 204

12.3.7 Manual Transport 205
12.3.8 Keyboard Shortcuts 205
12.3.9 Remote Control from Smart Panel 206
12.3.10 Session and Sign-in 206
12.4 Setup and Administration of Prompter 207
12.4.1 Prerequisites 207
12.4.2 Installation 207
12.4.3 Opening the Prompter 207
12.4.4 Connection to the Viz Mosart Server 207
12.4.5 Optional Server Setups 208
12.4.6 Session Timeout 208
12.4.7 Stored Settings 209
12.4.8 Remote Control 209
12.4.9 Troubleshooting 209
13 Named Overlays Editor 211
13.1 Key Features 211
13.2 Typical Workflow 211
13.3 Working with NOE 213
13.3.1 Overview 213
13.3.2 Session and Sign-in 214
13.3.3 The Overlay Table 214
13.3.4 Adding Overlays 215
13.3.5 Editing an Overlay 216
13.3.6 Managing Overlays 217
13.3.7 NamedCG Actions 218
13.3.8 NCS Content Placeholders 218
13.3.9 XML Import and Export 219
13.3.10 Keyboard Shortcuts 219
13.4 Setup and Administration NOE 221
13.4.1 Prerequisites 221
13.4.2 Accessing the Editor 221
13.4.3 Authentication and Roles 222
13.4.4 Configuring the Viz Pilot Edge Plugin 224
13.4.5 Main and Backup Servers 224
13.4.6 Troubleshooting 225
14 Troubleshooting 227
14.1 Where to start 227

14.2 Web Apps Configuration Tool 227
14.2.1 The Configuration Tool is refused, or shows only Home 228
14.2.2 The Mosart Web Proxy Service is not running or is not functioning properly 229
14.2.3 The Mosart Web Applications Service is not running or is not functioning properly 229
14.2.4 The External Proxy Service is unreachable or not functioning properly 230
14.2.5 Unable to reach the Viz Mosart Server (main or backup) 231
14.3 Sign-in and permission issues 232
14.3.1 A save did not reach the server 234
14.4 Slow loading or a client limit in the live apps 235
14.5 Issues with Mosart Web Server Connecting to the Mosart Server 236
14.5.1 Mosart Remote Control Service (RCS) 236
14.5.2 Firewall Ports 236
14.6 Browser Logs 237
14.6.1 To investigate browser logs 237
15 Appendix 238
15.1 Elgato Stream Deck Plugin 239
15.1.1 Installation 240
15.1.2 Configuration 240
15.1.3 Actions 241
15.1.4 Status 242
15.2 Vizrt SSO (Keycloak) 244
15.2.1 Working with Vizrt SSO (Keycloak) 244

Mosart Web Applications are value-adding utilities and plugins that simplify working with Viz Mosart. They are *not* part of a standard Viz Mosart delivery.

Copyright © 2026 Vizrt

1 Prerequisites

- A running Viz Mosart system. Individual web apps depend on the Viz Mosart server supporting them, so for the Viz Mosart version required by your Mosart Web Apps release, see the *Mosart Web Apps Release Notes* at the Vizrt Documentation Center.
- To use *Showmaker*, you must have Viz Mosart 5.14.3 or later.

Copyright © 2026 Vizrt

2 Getting Started

1. Collect the latest software bundle from Vizrt's customer download site at https://download.vizrt.com/ by navigating to /products/VizMosart/Latest Version/WebApplications/.

> **Note**: If *upgrading*, make sure an existing instance of the Mosart Web Applications Configuration Tool is **not** open.

2. Run the downloaded file *MosartWebApplications.Bundle-n.n.n.n.exe*.
   The Web Applications that will be installed are listed.

> **Info**: The Installer places these services and applications on the target machine:
> 1. **Mosart Web Applications**: A Windows service that serves the Mosart Web Applications.
> 2. **Mosart Web Proxy**: Caddy2 bundled within the Mosart Web Apps installer, running as a Windows service.
> 3. **Mosart Web Applications Configuration Tool**: A desktop web application for configuring and launching the web applications.
> 4. **The set of Mosart Web Applications** listed on the Installer.

3. Click **Install**.
   Any previously installed version of the Mosart Web Applications is automatically upgraded. All configurations and preferences are retained and you complete the installation by clicking **Finish**.
4. After successful installation, the **Mosart Web Applications Configuration Tool** displays, together with a desktop shortcut for quick access.
5. From the navigation sidebar on the left, select one of the following:

Sidebar entry	What it is for
Home	The landing page, which is also the app launcher: one card per installed web app. On a multi-studio site the cards are grouped, those that open against the selected studio under *Apps for* that studio, and the rest under **Cross-studio**. A single-studio installation shows one flat list.
App Configuration	Per-app settings on five tabs: NRCS Plugin, Showmaker, Template Editor, Named Overlays Editor and Smart Panel.
Server Configuration	The Mosart server addresses, proxy, HTTPS, authentication and the shared integrations. HTTPS and sign-in are documented separately, under HTTPS and Certificates and Authentication.
Studios	Definitions of the studios on a multi-studio site.
Help & Resources	The User Guide, Release Notes, API Docs, the Stream Deck plugin and the Mosart Server settings editor.

Copyright © 2026 Vizrt

Sidebar entry	What it is for
Quick setup	Re-enters the guided first-run wizard, at the bottom of the sidebar.

- If your system is *already configured*, open **Home** and launch any of:
  Showmaker
  NRCS Plugin
  Rundown Viewer
  Smart Panel
  Prompter
  Named Overlays Editor
  Template Editor (Early Access). Every installed app has a card. An app that ships switched off shows an **ENABLE** button instead of launching, and its card is moved to the front of the list; the Template Editor is the one that does, and its ENABLE button takes you to the Template Editor tab of App Configuration.
- If this is your *first running* of the Configuration Tool, or you will start using a new Web App, you must make some one-time configurations,
  as described in section Configuration Tool. The quickest route on a fresh installation is the **Quick setup** wizard, which opens on first run.

Note: The **Timing Display** web app was removed in Mosart Web Apps 3.1. Its capabilities are provided by **Smart Panel**. See Timing Display for the migration notes.

2.1 User Guides

Each web app's own section of this guide is listed in the navigation on the left. Inside the Configuration Tool, **Help & Resources** opens the copy of this guide that shipped with your installation, together with the release notes for that version.

Tip: The onboard descriptions are often early drafts. For *latest documentation*, visit the online resources at the Viz Mosart Documentation Center.

Copyright © 2026 Vizrt

3 Related Documents

- *Viz Mosart User Guide*: Operating Viz Mosart in a live production.
- *Viz Mosart Administrator Guide*: Configuring and customizing Viz Mosart and pre-production show design.

For more information about all of the Vizrt products, visit:

- www.vizrt.com
- Vizrt Documentation Center
- Vizrt Training Center
- Vizrt Forum

Copyright © 2026 Vizrt

4 Feedback and Suggestions

We encourage suggestions and feedback about our products and documentation. To give feedback and/or suggestions, please contact your local Vizrt customer support team at www.vizrt.com.

Copyright © 2026 Vizrt

5 Configuration Tool

The Configuration Tool enables rapid setup and management of all your Mosart Web Applications. This section presents advanced, mostly one-off configurations and references for the Mosart system administrator.

On *first run*, you must perform the one-time setups for the Mosart Web Apps. The quickest route is the **Quick setup** wizard, which walks you through the essentials. Everything it sets, and much more, remains available from the navigation sidebar.

The sidebar has four configuration entries, then a **Help & Resources** section, and **Quick setup** pinned at the bottom:

- **Home**: the app launcher, with one card per web app. See *Home* below.
- App Configuration: per-application settings, on five tabs in this order: **NRCS Plugin**, **Showmaker**, **Template Editor**, **Named Overlays Editor** and **Smart Panel**.
- Server Configuration: the proxy setup method, HTTPS and certificates, sign-in, the proxy web server address, the Mosart API access key, clip labelling, and the TriCaster, Media Sequencer, Viz Arc and Pilot Data Server integrations.
- Studios: defining production environments for a multi-studio site, each with its own Viz Mosart server and an optional backup.

Two further pages in this section go deeper into what **Server Configuration** exposes: HTTPS and Certificates, covering the proxy setup methods, the built-in certificate authority, custom certificates and their requirements; and Authentication, covering OpenID Connect sign-in and the user roles for the web apps.

5.1 Opening the Configuration Tool

The Configuration Tool is a web page served by the Mosart Web Apps server, at `/configtool` on the Web Apps address. On the server, the **Vizrt Mosart Web Applications Configuration Tool** shortcuts, on the desktop and in the **Vizrt** folder of the Start menu, open it in a window of its own. The window runs with administrator rights when your Windows account has them, and starting a shortcut again brings the open window forward rather than opening a second one. Apps you launch from the window open in your default browser.

When the window cannot run, the shortcuts say why, and no browser opens by itself. Without the Microsoft Edge WebView2 Runtime, a message says the tool needs it, with **Download page**; the Web Apps bundle installer includes the runtime. When the window fails to start, a message asks you to repair the runtime under **Settings > Apps > Installed apps**. With HTTPS and sign-in on, both messages also offer **Open in browser**, where someone with the `configtool-admin` role can change the settings. Starting a shortcut while the window is already open or still closing says so; try again in a moment. For a certificate this computer does not trust, see *Certificate warnings* below.

Upgrading to this version removes the earlier desktop Configuration Tool. The shortcuts keep their names and places, and open the new window.

5.1.1 Who can open it, and who can change settings

- **Reaching the tool**: by default the tool answers only on the Web Apps server itself, and a browser on another computer is refused. To open it from other computers, turn on **Serve Config Tool on the web** on Server Configuration, preferably together with HTTPS and sign-in.

Copyright © 2026 Vizrt

- **With sign-in off**, anyone who can reach the tool can use **Home**, but only the Configuration Tool window on the server, run by a Windows administrator of that machine, can change settings. A browser, even one on the server, gets **Home** and **Help & Resources** only.
- **With sign-in on**, the tool asks you to sign in like the other web apps, and changing settings needs the `configtool-admin` role. A signed-in user without it gets **Home** and **Help & Resources** (**User Guide**, **Release Notes**, **API Docs** and the **Stream Deck Plugin** download): they can launch the apps, see whether the services are up, and read the guides, but cannot change settings. The window on the server, run by an administrator, can still change settings without signing in, which is the way back in if a sign-in setting is wrong. See Authentication.

Without the right to change settings, the configuration pages, **Quick setup** and **Mosart Server settings** are dimmed, and Home says how to get that right.

Messages about the services and the settings appear in one status belt across the top of the window, on every page: a service that is down or restarting, an invalid channel templates folder, the Mosart Server out of reach, settings that could not be read, or a proxy that kept its previous configuration. Red rows are errors and yellow rows are warnings. A row offers the page whose settings fix it, **Web server log** and **Proxy server log**, and **Restart service** when a service is down. Most rows can be closed; a row opens again with the next check or save, or when it has something else to say. During Quick setup the belt leaves out the row about the services, the channel templates folder and the Mosart Server, because the wizard shows its own connection status.

5.1.2 The Configuration Tool window

Because it runs on the server, outside the Web Apps server, the window does what a browser cannot:

- It opens the tool at the server's own address. With the built-in proxy, it tries the proxy's address first, then the web server address, then `localhost`; with a manual proxy, the web server address, then `localhost`.
- Closing the window while it is starting, stopping or taking over the service asks first, as closing then may leave the service stopped (**Close anyway?**, with **No** as the default). Going back or forward, with Alt+Left or the mouse's back button, does nothing in the window.
- Opening it starts a stopped *Mosart Web Applications* service, and a stopped *Mosart Web Proxy* service when the bundled proxy is in use.
- The **Services and folders** button in the title bar, to the left of **Minimize**, opens a menu with **Restart Mosart Web Applications**, **Restart Mosart Web Proxy**, **Open web server log files**, **Open proxy server logs**, **Open certificate folder**, **Open in browser** and **Reload**. The two restarts need administrator rights. **Open in browser** opens the tool in your default browser.
- On the pages, a window run as administrator adds **Choose file** and **Choose folder** dialogs for the certificate, the private key and the templates folder, and a **Restart service** that works even when the Web Apps server does not answer. Every window adds **Open folder** for the built-in authority's root certificate and log links that open the log folders.

Certificate warnings

With HTTPS and a custom certificate, the window accepts the server's certificate even when this computer does not trust it, when it names another address, or when it has expired, provided it is exactly the certificate the settings name and the Mosart Web Applications service on this computer is the one answering. The status belt then warns, for example **The configured security certificate is not trusted by your computer's operating system.** If the

Copyright © 2026 Vizrt

certificate is not trusted, install it, or the authority that issued it, as trusted on this computer, then close the window and open it again. A certificate that names another address or has expired needs replacing; the warning stops once the new one is saved.

Any other certificate is refused, and the window shows a page of its own with the reason: not the certificate the settings name, another program answering at the web server's address (or that address is not this computer), a revoked certificate, or a web server with no certificate to serve because it cannot load the one the settings name. From there, **Stop the service and change the settings** opens settings-only mode, and except for a revoked certificate or none served, **Open in browser** shows the tool in your browser with its own certificate warning. When the window stopped the service for this and nothing has started it since, closing the window starts it again.

When the server refuses a save, a red row in the status belt reads **Not saved:** followed by the server's reason, such as a port already in use or a certificate that could not be loaded.

Settings-only mode

When the Mosart Web Applications service will not start, typically because of a setting, the window still opens the tool, in settings-only mode, so that you can correct it. The status belt gives the reason, with the last error the service logged, followed by **Settings-only mode: change the setting that stops it and save, which starts the service.** It offers **Web server log**, **Proxy server log** and **Start the service**, and the web apps cannot be launched meanwhile. Change the setting and save: the window starts the service and opens the tool there, on the page you saved from. **Start the service** starts it without a change. If the service still does not start, the window comes back to settings-only mode with the new reason. **Open in browser** is not available in this mode.

If even that cannot run, the window shows a page of its own: the state of both services with a **Start** or **Restart** button for each, two fixes for the settings that most often stop the server, **Turn HTTPS and sign-in off** and **Reset the web server ports** (back to 65142 and 65143), **Try again**, and the last lines of the web server log. Each fix keeps a copy of the settings file as it was, then starts the service.

> **Info**: Starting services and changing settings from the window needs a Windows administrator. Only one window at a time can take the service over, for settings-only mode or one of the fixes: an administrator in another Windows session, over Remote Desktop for example, is told that another window is already working on the service.

5.2 Saving your changes

This applies to every page in this section. **Save** and **Discard changes** appear at the bottom of a page only while that page has unsaved edits. If you try to leave a page or tab with unsaved edits, a **Save changes** dialog offers **Save**, **Discard** or **Cancel**, and the move happens only once you have chosen. Most changes apply as soon as you save, with no restart: the Web Apps server reads the new settings straight away, and the bundled web proxy reloads its configuration. Three kinds of change still restart the Web Apps server: turning HTTPS on or off, turning sign-in on or off, and changing where the templates are stored (**Location** on the **NRCS Plugin** tab, or the database details while the templates are in a database). The tool restarts the server itself, showing **Restarting services ...** until it answers again. A save that changes the address the tool is opened on, such as turning HTTPS on or off or changing that port, reopens the tool at its new address, on the page you saved from. A save that changes the sign-in settings reloads the tool, so that you sign in again with the new settings.

Copyright © 2026 Vizrt

5.3 Home

Home is the app launcher. Each web app gets a card showing its logo and its name, and clicking the card launches the app.

- An app that ships switched off shows an **ENABLE** button instead of launching. The Template Editor is the one that does, and its **ENABLE** button takes you straight to the **Template Editor** tab of **App Configuration**. That card also carries an early-access note. Without the right to change settings, the card reads **Not enabled** instead.
- Cards still waiting to be enabled are promoted towards the front of the list, so what needs attention is what you see first.
- When more than one studio is defined, a studio strip appears at the top and the cards split into two groups, *Apps for "<studio>"* and **Cross-studio**. A single-studio installation keeps one flat list.

5.4 Quick setup

On first run the Configuration Tool opens a guided **Web Apps quick setup** wizard that collects the minimum needed to get the Web Apps talking to Viz Mosart. It detects the machine it is running on and pre-fills what it can.

It opens on a welcome screen offering **Get started** and **Skip setup**. The steps are:

1. **Access** (*How will people reach the web apps?*): the **Web server address** and the **Browser port**. The address starts as the one already saved, unless that is `localhost` or another address only this machine can use; then it starts as the machine's name, which other computers can reach. The step does not accept `localhost` or another address only this machine can use, nor an address with `http://` or a port. **Activate** makes this address the proxy's as well. This is also where you choose between **Plain HTTP** and **Secure with HTTPS**.
2. **Security** (*Certificates and sign-in*): shown only when you chose **Secure with HTTPS** on the Access step. Offers **Automatic (built-in CA)** or **Custom certificate**, and, with a custom certificate, a **Require sign-in (OpenID Connect)** toggle with **Issuer URL**, **Client ID**, **Secret** and **Scopes (optional)**.
3. **Mosart server** (*Connect your Mosart server*): a **Studio name**, the Mosart server address, an optional backup server address, and the API access key.
4. **Review & activate**: the screen is headed **Ready to go live** and reviews everything you entered, with **Activate** to apply it.

A rail on the left tracks your progress. You can jump back to any step you have already reached, but you cannot skip ahead, because each step is validated as you continue.

A **System overview** panel on the right builds up as you make each choice, showing the chain from users, through the Web Apps server, to the Mosart server.

While the settings are saved, the button reads **Activating…**, and **Back** and the steps on the rail wait. The final screen appears only once the server accepts the settings; if it refuses them, the wizard stays on **Ready to go live** and the status belt says why in a **Not saved:** row. After **Activate**, the final screen reports whether the apps are reachable at the address it names, headed **Your web apps are live** or **Setup saved, but not reachable yet**. **Go to Home** takes you into the tool and **Run setup again** restarts the wizard. If you chose **Automatic (built-in CA)**, that screen also reminds you that browsers keep warning until the root certificate is trusted on client machines, and gives the root certificate's path on the server, with **Open folder** in the Configuration Tool window, or **Copy path** in a browser.

Copyright © 2026 Vizrt

Info: You do not have to complete the wizard on first run. If you leave it, it is not forced open again the next time you start the tool, and you can re-enter it at any time from **Quick setup** at the bottom of the navigation sidebar.

5.5 Help & Resources

A **Help & Resources** section at the bottom of the navigation sidebar gathers documentation and tools:

Item	What it opens
User Guide	This documentation, on a page inside the Configuration Tool, served at `/docs` on the Web Apps address. The trailing button opens the same page in your default browser.
Release Notes	The Mosart Web Apps release notes for this version, at `/releasenotes` on the Web Apps address.
API Docs	The Mosart REST API documentation (Swagger), at `/mosart/swagger/index.html` on the Web Apps address. It is proxied through to the Viz Mosart server's Remote Control Service, so it always describes the currently active server, main or backup. In a multi-studio installation the proxy also exposes a per-studio path, `/mosart/<studio>/swagger`.
Stream Deck Plugin	Downloads the bundled plugin file `com.vizrt.mosart-remote.streamDeckPlugin`, ready to install on the Stream Deck. Nothing is launched. The hover text on its trailing button is **Download plugin**. See Elgato Stream Deck Plugin.
Mosart Server settings	The settings editor served by the Mosart server itself, so you can reach Manus Administrator settings without leaving the Configuration Tool. It opens the installation's **main** Viz Mosart server at that server's own address, not through the web proxy.

Why the API documentation works through the proxy. The proxy tells the Viz Mosart server which prefix it sits behind (`/mosart`, or `/mosart/<studio>` for a studio), so the Swagger page's own *try it* calls go back through the Web Apps address instead of straight at the Mosart server. That is what makes the API documentation usable from a browser that can only reach the Web Apps host. `/docs` and `/releasenotes` are served by the Mosart Web Apps server itself and are reachable without signing in.

Info: The tool probes the main Viz Mosart server continuously, and the **Mosart Server settings** row dims for four distinct reasons, each explained on hover:

Copyright © 2026 Vizrt

- You do not have the right to change settings. The hover text reads **Requires the configtool-admin role**.
- The settings editor is not enabled on that server. The hover text reads **RCS's Manus settings editor is off: in the Remote Controller Service settings, under Web editors, turn on "Enable Manus settings editor", then restart RCS.**
- OIDC authentication is enabled on the Mosart server, and the settings editor has no browser sign-in. The hover text reads **RCS sign-in is on, and RCS's settings editor has no browser sign-in: change the settings in Manus Administrator.**
- The Viz Mosart Remote Control Service is not reachable at all. Start it. The hover text reads **The Mosart Server can't be reached.**

If the Web Apps server does not trust the Mosart server's certificate, or the certificate does not match the Mosart server address, the row stays usable and its hover text warns: **Open the Manus Administrator settings editor served by the Viz Mosart Server. Your browser may warn about its certificate: the Web Apps server doesn't trust it, or it doesn't match the Mosart Server address.**

> **Info**: With OIDC authentication enabled on the Mosart server, **Mosart Server settings** cannot be used from a browser. Edit the settings on the Mosart server itself instead: in **Manus Administrator**, type *settings* in the console, or in **AV Automation** use **Devices > Settings**. That is also how you edit a backup server, or any other Mosart server. The Showmaker MOS settings can still be pushed to Manus Administrator from App Configuration, in a browser signed in with an account that has the `mosart-admin` role.

5.6 Troubleshooting

See section Troubleshooting.

Copyright © 2026 Vizrt

5.7 App Configuration

Settings here are needed when you use the NRCS Plugin or Showmaker for the first time, or have installed the Mosart Web Apps bundle on *a separate server* to the Viz Mosart server.

> **Note**: *Showmaker* always requires the NRCS Plugin to be correctly configured.

- NRCS Plugin
  - To configure the NRCS Plugin
  - Template type filter order
  - Show Add to Story buttons
- Showmaker
  - Prerequisites
  - Connection settings
    - To establish communication between a single MOS device and Showmaker
    - Default values and multiple MOS devices (main/backup or multiple studios)
  - Feature settings
  - Assets integrations settings
  - Prompter integration settings
- Named Overlays Editor
- Template Editor (Early Access)
- Smart Panel
  - Time format
  - Time zone
  - Panel access

5.7.1 NRCS Plugin

To configure the NRCS Plugin

1. First perform the Getting Started (Installation) procedure.
2. Select **App Configuration** and click **NRCS Plugin**.
3. From the **NRCS type** drop-down menu, select your newsroom system type. The options are **Showmaker**, **iNews**, **ENPS**, **Octopus**, **OpenMedia** and **Other**. Showmaker is in the list because Showmaker itself can be the newsroom system.
4. Enter **Mos ID**. The field is shown for every NRCS type except **iNews**, so it applies to Showmaker, ENPS, Octopus, OpenMedia and Other. Its help text says the field is required when the newsroom system uses the MOS protocol. The tool does not block a save when it is left empty, so the value has to be right by inspection.

> **Tip**: To find your Mos ID, in the Manus Administrator console, type *settings*. For more details, see the *Viz Mosart Administrator Guide*, section *Manus Administrator > Settings Editor MOS > NRCS Configuration*.

5. Back in the Web Apps Configuration Tool **App Configuration** page for the NRCS Plugin, in section **ACCESS MOSART TEMPLATES**, specify the location of your Viz Mosart templates. **Location** offers two options:

Copyright © 2026 Vizrt

   a. **Local**: a folder on the Web Apps server, or a network path the server can reach. The default is `C:\channeltemplates`. Type the path as the server sees it, whichever computer you are using the tool from, and make sure that folder has adequate permissions. In the Configuration Tool window on the server, run as administrator, **Choose folder** opens a folder dialog instead. The tool scans for three files, `channeltemplates.xml`, `avconfig.xml` and `newsroomsettings.xml`, and reports the folder as valid or invalid as you type.
   b. **Database**: if template details are stored in a database, enter host access and authentication details.

   A save is refused while the folder is invalid, which is the one validation this tab enforces.
6. Underneath the fields, **CONFIGURE NRCS PLUGIN** shows a **Preview URL** with a copy button beside it. That is the URL to hand to the newsroom system: the Web Apps address plus `/plugin`, with `nrcs=<type>` appended for every type except **Other**, and `mosid=<value>` for every type except **iNews**. Click the link to open it in your default browser and confirm the plugin loads before you configure the newsroom system.
7. Click **Save**.

Template type filter order

The NRCS Plugin shows a row of filter buttons so journalists can narrow the template list to one type. The control that governs them is **Template type filter order**, under **UI SETTINGS** on the **NRCS Plugin** tab.

It lists all thirteen types, namely Camera, Clips, Voice over, Live, Graphics, DVE, Jingle, Telephone, Adlib, Break, Video wall, Sound and Accessories, and you reorder them by dragging a row or using the up and down arrows. Put the types your journalists use most at the top.

When the templates come from a local folder and its `newsroomsettings.xml` names an alias for a type, that row is labelled with your newsroom's own name instead of the name above. Types the file does not name keep the names above, as do installations that store the templates in a database, where there is no file to read.

> **Note**: This sets the **order only**. There is no way to hide a template type: any type missing from the saved order is appended in default order, so every type always reaches the plugin. The order applies to everyone using the plugin, and the **ALL** filter always stays first.

Show Add to Story buttons

Show Add to Story buttons, under **UI SETTINGS** on the **NRCS Plugin** tab, is on by default. Turn it off when journalists should place templates in a story by drag and drop only: the plugin then hides its **Add to Story** buttons, and **Enter** and double-click no longer add a template. Items opened from the story can still be updated or replaced. In iNEWS, its own **OK** and **Apply** buttons still add a template.

The setting applies to everyone using the NRCS Plugin, the next time the plugin is opened. This includes the NRCS Plugin panel in Showmaker, which then hides its **Add to Story** and **Add to show assets** buttons too. Click **Save** to apply it.

> **Info**: To complete the NRCS Plugin one-time setup, you will also need to visit the **Server Configuration** page.

Copyright © 2026 Vizrt

5.7.2 Showmaker

Prerequisites

- A running Viz Mosart 5.14.3 or later.
- *Manus Administrator* is running.
- The *Viz Mosart client UI* is running and can display a rundown.

If either Manus Administrator or Viz Mosart is down, start them now. You only need to do this once after a server reboot.

> **Info**: Users *can* run Showmaker standalone without a connected Viz Mosart. This user scenario would be for early versions of a rundown (running order) where structure is being drafted, rather than operations verified.

Connection settings

Showmaker uses standard MOS protocol. The **Connection settings** heading on the Showmaker tab covers two blocks, **MOS COMMUNICATION SETTINGS** and **MOS DEVICES**.

A fresh installation has *one* MOS device pointing to your Mosart server: a single row in the **MOS DEVICES** table, whose columns are **ID**, **URL** and **Studio**. While the table has exactly one device, the row also carries the control for handing the matching settings over to Manus Administrator, described below.

The add and edit dialog labels the same three fields **ID ***, **URL** and **Studio ***. A new device pre-fills **Studio** with *Default*, and the **URL** is pre-filled with the local Mosart address only for the very first device.

To establish communication between a single MOS device and Showmaker

The Configuration Tool reads the Manus Administrator settings from the main Viz Mosart server, wherever Showmaker runs. There are two procedures, depending on whether it can read them.

The Configuration Tool can read the settings from the main Viz Mosart server

When the settings can be read, the Configuration Tool can write the matching MOS configuration into Viz Mosart for you.

1. From the **App Configuration** page, select *Showmaker*.
2. Review the values in **MOS COMMUNICATION SETTINGS**. If necessary, adjust the defaults by referring to the table below.
3. Click the icon button in the last column of the MOS device row. It appears only while the table holds exactly one device. Its tooltip reads **Press the button to set up the communication with Viz Showmaker.** With changes to make, it opens **Inspect changes and push settings to Manus Administrator**, showing current and new values side by side. The settings are read again each time the dialog opens, so it reflects any change made on the Mosart server since you opened the tool. If the settings already match, or Manus Administrator has no NCS type set yet, it opens **Current Manus Administrator settings** instead, with no **Confirm**. In the second case it lists the values under **Set these settings manually**: enter them in Manus Administrator as described below.
4. The dialog notes that the settings are applied to Manus Administrator live, so it does not need to be stopped.
5. To complete the action, click **Confirm**. **Settings applied to Manus Administrator.** confirms the push. If it fails, the message reads **Failed to apply new settings.** followed by the reason, as listed below.

Copyright © 2026 Vizrt

Info: Before anything is written, a backup of the current Manus Administrator settings is created at *C:\ProgramData\Mosart Medialab\ConfigurationFiles*, so the previous configuration can be restored.

The Configuration Tool cannot read the settings

When the settings cannot be read from the main Viz Mosart server, set Viz Mosart up manually for connecting to Showmaker:

1. Review the values in **MOS COMMUNICATION SETTINGS** and, if necessary, adjust the defaults as presented in the table below.
2. Click the icon button in the last column of the MOS device row. Its tooltip gives the reason the settings could not be read, followed by **Press the button to view the settings to enter in Manus Administrator.** It opens the **Settings for Manus Administrator** dialog, which lists the values to enter, under a warning that says why they could not be read. **Confirm** is dimmed, with the reason as its tooltip, and **Close** closes the dialog. If the reason is one you can fix (see below), fix it and click the icon button again: the settings are read again, and you can push them as described above.
3. On the Viz Mosart server, in the Viz Mosart Manus Administrator console, type *settings*.
4. From the **Settings** menu that appears, navigate to the section **MOS**.
5. Configure the MOS settings as presented in the **Settings for Manus Administrator** dialog in the Web Apps Configuration Tool.
6. Click **Save**.
7. (optional) If changes do not appear immediately, restart the Windows service *Mosart Web Applications* (service name *MosartWebApplications*).

When the settings cannot be read or pushed

The Configuration Tool says why. The same message appears in the warning of the **Settings for Manus Administrator** dialog, on its dimmed **Confirm**, when you hover the icon button, and after **Failed to apply new settings.**

Message	What to do
RCS's Manus settings editor is off: in the Remote Controller Service settings, under Web editors, turn on "Enable Manus settings editor", then restart RCS.	On the main Viz Mosart server, turn on the setting the message names and restart the Remote Control Service. The Configuration Tool reads and pushes the settings through that editor.
The Mosart Server can't be reached.	Check that the Remote Control Service is running on the main Viz Mosart server, and that the Web Apps server can reach it.
The Mosart Server refused the request (HTTP *n*).	The Mosart server answered, but turned the request down for a reason the Configuration Tool cannot name. *n* is the status it answered with.
The Mosart Server uses HTTPS, but the Web Apps reach it over http: turn on HTTPS with a custom certificate in Server Configuration, as HTTPS must be on for both.	Turn on HTTPS with a custom certificate on Server Configuration. See HTTPS and Certificates.

Copyright © 2026 Vizrt

Message	What to do
The Mosart Server's Manus settings file (manus_settings.json) is missing or can't be read: start Manus Administrator once on the Mosart Server, then try again.	Start Manus Administrator once on the main Viz Mosart server. It creates the file on its first start.
The Web Apps server doesn't trust the Mosart Server's certificate, or it doesn't match the Mosart Server address.	Make the Web Apps server trust the Mosart server's certificate, and check that the certificate is issued for the Mosart server address the Web Apps use. See HTTPS and Certificates.
RCS sign-in is on: sign in to the web apps with an account that has the mosart-admin role.	OIDC authentication is enabled on the Mosart server. Sign in to the Configuration Tool in a browser with an account that has the `mosart-admin` role.
Your account doesn't have RCS's mosart-admin role.	Have the `mosart-admin` role assigned to your account in your authentication provider.
RCS sign-in is on, but the web apps don't sign people in: turn on sign-in for the web apps, then sign in with an account that has the mosart-admin role.	Enable authentication for the web apps (see Authentication), then sign in with an account that has the `mosart-admin` role.
RCS sign-in is on, and the Config Tool app can't sign in to RCS: to push these settings, open the Config Tool in a browser and sign in with a mosart-admin account.	Shown in the Configuration Tool window on the Web Apps server. Open the Configuration Tool in a browser, sign in with an account that has the `mosart-admin` role, and push the settings from there.
RCS sign-in is on, but the web apps don't sign people in, and the Config Tool app can't sign in to RCS: turn on sign-in for the web apps, then push these settings from a browser signed in with a mosart-admin account.	Shown in the Configuration Tool window on the Web Apps server. Enable authentication for the web apps, then push the settings from a browser signed in with an account that has the `mosart-admin` role.

Info: With OIDC authentication enabled on the Mosart server, push the settings from the Configuration Tool opened in a browser, signed in with an account that has the `mosart-admin` role as well as `configtool-admin`. Authentication must be enabled for the web apps too, which needs HTTPS with a custom certificate. The Configuration Tool window on the Web Apps server cannot pass a sign-in on to the Mosart server. See Authentication.

Copyright © 2026 Vizrt

Default values and multiple MOS devices (main/backup or multiple studios)

Table Property	Description
NRCS ID	This is the Newsroom System ID for Showmaker. Set this ID on any MOS-enabled NRCS device to establish communication with Showmaker. For Viz Mosart, this value must be set in Viz Mosart Manus Admin. Default: *SHOWMAKER.MOS*.
Add MOS Device	For defining multiple MOS devices. Click **Add MOS Device** and follow the guidelines in this table.
ID	This is the ID of a MOS device configured to communicate with Showmaker. For Viz Mosart, copy this value from **Manus Admin console > Type** *settings* **> MOS** section. Default: *MOSART.MOS.SHOWMAKER*.
URL	The URL for connecting to a MOS device. For a Viz Mosart server, the URL is *ws://mosart-server-host:10540/mos/mosartsocket*. The port in the URL (*10540*) is the MOS lower port configured in the **Settings** menu of Manus Admin, under the **MOS** section. Default: *ws://localhost:10540/mos/mosartsocket*.
Studio	The Studio label that this device belongs to. Type a new name to create a Studio. A *Studio* is a label that groups one or more MOS devices, representing one production environment: typically a main Viz Mosart server with an optional backup, or a single gallery in a multi-studio site. The Studio exists as long as at least one MOS device references it, and disappears when no device uses the name anymore. Once Studios are configured, each show (and optionally each rundown) can be assigned to a Studio, and that assignment governs which Mosart server or servers the rundown is sent to. **Failover without downtime**: main and backup Mosart servers under one Studio means a failover keeps the same Showmaker rundowns on air. **Per-rundown overrides**: a show defaults to one Studio, but individual rundowns can be reassigned, shown with an orange badge in the UI, which is useful for rehearsals or last-minute changes.
Examples	Typical configurations: **One studio with main + backup**: two rows with the same Studio label and the same MOS ID, and different URLs (`studio1-main` and `studio1-backup`). **Multiple studios**: one or more rows per Studio. Each Studio can have its own main and backup pair, with the same or different MOS IDs. **Note**: the same MOS device URL cannot be used twice across the table. Duplicates are rejected with an error message.

Feature settings

Story Recorder
Story Recorder lets operators mark stories for recording in Viz Mosart, optionally publish the resulting clips to

Copyright © 2026 Vizrt

online platforms (only available with TriCaster) or replace the original story script with the recorded version. To make these controls available in Showmaker:

1. Configure Story Recorder in Viz Mosart first. See the Vizrt Documentation Center, under *Viz Mosart Administrator Guide* > *Operational Examples* > *Story Recorder Setup*.
2. In the Configuration Tool, under App Configuration, select **Showmaker**, scroll down to **Feature settings** and turn on **Enable Story Recorder controls**.

Once enabled, every story in a rundown gains a record control; stories in a rundown template do not. The publish action and the **Use recording** toggle appear on a story only once that story has a recording. With authentication enabled, only a user with the showmaker-admin role can operate these controls: a viewer sees the record state and the **Use recording** toggle but cannot change them, and the publish action is not shown. For how operators use these controls, see Working with Showmaker.

Assets integrations settings

If you use either Viz Pilot or Viz Flowics to provide assets for your show, you need to configure some one-time settings. The integrations are displayed in the Showmaker integrations sidebar together with the Mosart web app, *NRCS Plugin*.

> **Info**: These integrations appear in the Showmaker integrations sidebar as selectable, embedded MOS plugins (iFrames). They communicate directly with Showmaker in the browser using MOS messages, so no additional server-side configuration is required beyond specifying the URL and MOS Plugin ID.

Viz Pilot: to enable Viz Pilot graphics in the Assets panel

1. If you will run Showmaker with assets from Viz Pilot, select the checkbox for **VIZ PILOT GRAPHICS AND ELEMENTS**.
2. By following the on-screen information icon, provide values for
   a. **Application URL**: Where Viz Pilot Edge is running. Default: *http://localhost:8177/app/pilotedge/pilotedge.html?mosid=pilot*. Enter the host of your Viz Pilot Edge installation, if not on the same machine.
   b. **MOS Plugin ID**: The MOS ID that Viz Pilot Edge announces. Default: *PILOT*. This value must match what your NRCS / Mosart side anticipates.
3. Click **Save**.

Viz Flowics: to enable Viz Flowics graphics

1. Select the checkbox for **VIZ FLOWICS GRAPHICS** and ensure you have your Viz Flowics (cloud application) credentials to hand.
2. By following the on-screen information icon, provide values for
   a. **Integration token**: A token issued by Flowics support. It is stored encrypted and displays as `**********` after saving.
   b. **Application URL**: Where Viz Flowics is running.
   c. **MOS Plugin ID**: Default: *flowics.com.mos*. Change only if your NRCS / Mosart side anticipates a different id.
3. Click **Save**. Viz Flowics now appears as a tab in the Showmaker Assets panel.

Prompter integration settings

Below the asset integrations, **Prompter integration settings** carries a **LEGACY MOS 2 PROMPTER (SOCKET)** checkbox. Enable it and four fields appear: **Prompter host / IP**, **Lower port**, **Upper port** and **Prompter MOS ID**. Showmaker then dials out to a prompter that speaks MOS 2 over TCP and pushes the running order and script to it.

Copyright © 2026 Vizrt

Field	Default
Prompter host / IP	`127.0.0.1`
Lower port	`10540`
Upper port	`10541`
Prompter MOS ID	`ForPrompt`

Note: On the prompter's own MOS settings, set its remote NCS ID to the **NRCS ID** configured above, and use the same **Prompter MOS ID**. Only rundowns activated for a MOS device are sent.

Info: To complete the Showmaker one-time setup you will need two other pages: **Studios**, for the Viz Mosart server and optional backup that Showmaker's rundowns are sent to, and **Server Configuration**, for how browsers reach the Web Apps. The Mosart main and backup server fields are no longer on Server Configuration; they live on Studios.

5.7.3 Named Overlays Editor

The Named Overlays Editor has its own tab in **App Configuration**, with a single block, **Assets integrations settings**. It carries a **VIZ PILOT GRAPHICS AND ELEMENTS** checkbox and two fields, **Application URL** and **MOS Plugin ID**. There is no display name to set: the plugin is always registered as Pilot Edge.

The first time you open this tab, the tool registers a Pilot Edge entry for you. It copies the Showmaker tab's Pilot Edge settings if one is configured there, and otherwise uses the defaults *http://localhost:8177/app/pilotedge/pilotedge.html?mosid=pilot&sendbutton* with MOS Plugin ID **PILOT**.

Clearing the checkbox is what leaves the editor without a provider panel. It then still works for creating custom overlays by hand and for XML import and export. For everything else about the app, see Setup and Administration NOE.

5.7.4 Template Editor (Early Access)

The **Template Editor** tab is where the app is switched on. The Template Editor ships switched off, and **ENABLE TEMPLATE EDITOR** at the bottom of the tab is what turns it on. Above the switch, the tab states the three things to know first:

- **Explore and test**: this is an early-access build, and it is not supported for production use.
- **Backup**: it shares the NRCS Plugin's template storage and supports file-store templates only, so back up your channel templates XML before you start.
- **Control access**: as a web app it is reachable by everyone on the network unless authentication is configured.

For the full procedure, including what happens on save and how to open the app, see Setup and Administration of TE.

Copyright © 2026 Vizrt

5.7.5 Smart Panel

The **Smart Panel** tab is the last of the five, and it governs how Smart Panel writes its two clock values, **Time of Day** and **Next Break Time**, and which devices can run a panel. Smart Panel reads the clock settings once, when it starts, so reload any panel that is already open after you save.

Time format

The **TIME FORMAT** block decides how those two values are written. **Use browser default** is selected out of the box: the clocks then follow the language settings of the browser on each Smart Panel machine, always with a two-digit hour. A **Preview** beside the checkbox shows the current time written the way the panels will write it, so you can check a choice before saving.

Clear **Use browser default** to set the format explicitly. This is what you need when the operator's Windows regional format differs from the browser's language, because a browser cannot read the Windows *Regional format* setting. Four controls appear:

- **Regional format**: a searchable list of the regional formats the Web Apps server offers, the same list Windows offers as regional formats. The choice decides the separators and the AM and PM text.
- **24-hour** or **12-hour**. The default is **24-hour**.
- **Leading zero**: on by default, so the hour is always written with two digits.
- **Use this browser's format**: fills the three controls above from the time format of the browser you are using the tool in, which follows that browser's language. In the Configuration Tool window on the server, run as administrator, the button reads **Use this PC's settings** instead and takes the Windows regional format of the user signed in on the server, which saves finding it in the list.

Time zone

The **TIME ZONE** block decides which clock the two values follow:

- **Local (browser)**, the time zone of each Smart Panel machine. This is the default, and it is what you want when studios or control rooms sit in a different time zone from the Viz Mosart server, so a presenter's clock reads the time where they are.
- **Mosart server**, the time zone of the Viz Mosart server. Choose this when everyone should read the same clock as the automation system, wherever the panel is opened.

> ⚠ **Note**: These settings need a matching Viz Mosart server version. Against an older server the clocks keep their previous behaviour, showing the Mosart server's wall clock in that machine's own 12 or 24 hour format.

For what an operator sees, see the Timer section of Working with Smart Panel.

Panel access

A Smart Panel link carries no Mosart key. Where Mosart Web Apps runs without OIDC sign-in, each device that runs a panel is instead paired once with a code, and the Web Apps server adds the Mosart REST API key to that device's button presses itself, using the **Authorization Key** on Server Configuration. A signed-in panel on an OIDC site needs no pairing: its own sign-in authorizes its button presses. A panel link can therefore be shared freely. The **Authorization Key** field appears on Server Configuration only with **Automatic proxy setup**. A site with **Manual**

Copyright © 2026 Vizrt

proxy setup can pair devices only if the key was saved while it still used **Automatic proxy setup**, and its own proxy sends the Smart Panel command and status routes to the Web Apps server, as the Smart Panel setup page describes. Otherwise it keeps opening panels with the key in the link for now. The **PANEL ACCESS** block is where you pair and remove devices. Unlike the rest of the tab, it acts at once, with no **Save**.

1. Open the panel on the new device. The first time, it asks for a pairing code.
2. On the **Smart Panel** tab, click **Show a pairing code**. On a multi-studio installation, first choose the studio the device is for: the device joins that studio and cannot choose another.
3. Read the six digits out to the person at the panel, who enters them. The code works for ten minutes and for one device, and the tab shows how long it has left. To send it in a message instead, click **Copy code** beside it. Should the browser refuse to copy, the tab says so; read the code from the screen.

Once the device has paired, the code disappears and the device is listed under **DEVICES THAT CAN RUN A PANEL**, with when it was paired and last used, and on a multi-studio installation its studio. **Generate new** replaces the code with a fresh one, and **Hide the code** withdraws it. Leaving the tab withdraws it too, and so do ten wrong entries.

Remove on a device's row, confirmed in **Remove this device?**, stops that one device: its next button press is refused. No key changes, so every other panel, Stream Deck and script carries on, and the device can pair again with a new code.

> **Info**: Until a device has paired, panels can still be built and viewed on it, but their buttons stay locked.

Copyright © 2026 Vizrt

5.8 Studios

A multi-studio site runs one central Mosart Web Apps host against several Viz Mosart back ends. The **Studios** page in the navigation sidebar is where you define them.

> ⚠ **Note**: Multi-studio support is being delivered in stages. The applications differ in how far they support it today, so check Studio support per application below when planning a multi-studio installation.

5.8.1 Defining studios

The page is headed **Studios** with a count of how many are defined, and describes itself as mapping each studio to a Viz Mosart server backend and an NRCS, so that studios become routable targets across the Web Apps. The table columns are a row number, **STUDIO**, **HOST**, **PORT**, **NRCS DEVICE** and **BACKUP**.

Each studio entry takes:

- **Studio name**: the label used throughout the Web Apps to identify this production environment.
- The Viz Mosart server **hostname or IP** for that studio, and its port.
- An optional **Backup server**, added with **Add backup** in the BACKUP column, which opens a **Backup server** sub-row beneath the studio with its own host and port. A plain shield icon marks a studio that has a backup, and a shield with a check marks the backup row itself.
- An **NRCS device**, selected from the devices you have configured.

Use **Add Studio** to create further entries. A new row arrives pre-named `studio-<n>`, pre-filled with the current default Mosart port and the first available NRCS device.

Studio names become URL slugs

On save, a studio's name is normalised to a URL-safe form: lower-cased, with every run of other characters turned into a single hyphen. *Studio 1* is stored and displayed as `studio-1`.

That slug is the studio's identity everywhere. The proxy routes it as `/mosart/<studio>/…`, the Home page launches studio-scoped apps with `?studioId=<studio>`, and it is the label on the Home page's studio strip. Two names that normalise to the same slug are rejected while you type, with **Name must be unique (collides with another studio)**. So choose names that read well in lower case with hyphens, and expect the tool to rewrite what you typed.

What the page validates

Entries are validated as you type, and the messages appear on the row itself:

- **Between studios**: two studios may not point at the same Viz Mosart backend, meaning the same **host and port** together. Two studios on the same host with different ports are allowed. The check resolves host names, so a short name, its FQDN, `localhost` and the machine's IP all count as the same machine.
- **Within one studio**: the rule is stricter. The backup must be on a **different host** from the main server, since a pair on one machine is no redundancy at all.

Copyright © 2026 Vizrt

- A host must be a hostname or IP address with no scheme, port or spaces, and a port must be a number from 1 to 65535.

Save is blocked while the conflict check is still running, so a clash cannot slip through by saving quickly.

With more than one studio defined, the Home page gains a studio selector: studio-scoped applications launch against the selected studio and connect to that studio's Viz Mosart server, while applications that work across studios are listed under **Cross-studio**. A single-studio installation keeps the flat layout.

Info: The Configuration Tool retargets the Mosart port on every studio row when you change certificate mode: **55168** when you supply your own certificate, **55167** otherwise, because the Viz Mosart servers serve HTTPS only with a custom certificate. It only moves a port that still holds one of those two defaults, so a port you typed yourself is left alone and has to be updated by hand. This applies to each backup port as well. See the HTTPS and Certificates page.

5.8.2 Studio support per application

Application	How it uses studios
Rundown Viewer	Launches per studio, and can switch studio from a picker in the timers bar.
Smart Panel	Launches per studio from the Home page. Panel layouts are shared by the installation.
Prompter	Launches per studio from the Home page.
Named Overlays Editor	Has its own studio selector in the header toolbar, and works against the selected studio's Mosart server.
Showmaker	Assigns each show, and optionally each rundown, to a studio, which decides where the rundown is sent. Its studio list comes from the Showmaker MOS devices in **App Configuration**; the **NRCS device** field on this page links a studio to that device. Showmaker is grouped under **Cross-studio** on the Home page because it works across studios, but launching it from Home still carries the selected studio: the tool follows that studio's **NRCS device** to the MOS device's Studio label and passes it, so Showmaker opens filtered to that studio's shows. A studio with no NRCS device set launches Showmaker unscoped, which is why the **NRCS DEVICE** column matters even on a single-studio installation.
NRCS Plugin, Template Editor	Work across studios, against the installation's template source.

Copyright © 2026 Vizrt

5.8.3 Editing a studio's Mosart server settings

Mosart Server settings under Help & Resources opens the settings of the installation's **main** Viz Mosart server only. A studio's own server, and any backup server, are not reachable from there.

To edit the settings of any other Mosart server, open the settings on that machine from a Viz Mosart desktop application: in **Manus Administrator**, type *settings* in the console, or in **AV Automation** use **Devices > Settings**. This is also the way to edit settings when **OIDC authentication is enabled** on the Mosart servers, because the browser-based editor has no sign-in of its own. See Configuration Tool.

> **Note**: Settings that must match across a studio's pair, HTTPS and authentication in particular, have to be applied on the main **and** the backup server separately, and the Mosart Remote Control Service restarted on each. A configuration that works until a failover is usually a backup server that was never updated.

Copyright © 2026 Vizrt

5.9 Server Configuration

Most settings here are needed when you install the Mosart Web Apps bundle on *a separate server* to the Viz Mosart server, move the Web Apps to a different server, or need to define the Viz Mosart backup server. For securing the connections, see **HTTPS and Certificates** and **Authentication**.

- Server addresses
    - Ports the tool manages
- Serve Config Tool on the web
- Sign-in and sessions
- Mosart API Access
- Media Sequencer
- Viz Arc
- Pilot Data Server
- Video Clips
- TriCaster Live Previews

5.9.1 Server addresses

Under the automatic proxy setup, all default settings are pre-filled and no further action should be necessary. If you need to change any values:

- **PROXY WEB SERVER, Address**: Hostname or IP address of the machine where the Mosart Web Applications are installed. The system attempts an auto-detect. If the *FQDN is required* (for example, by a certificate), provide this instead.

Under the manual proxy setup, where you run your own reverse proxy, this becomes a single **URL** instead of an address and port.

> **Info**: The *Viz Mosart* server addresses are not on this page. Each studio carries its own Viz Mosart host, port and optional backup server on the Studios page. If you are following older instructions that send you here for a main or backup Mosart server, go to Studios instead.

Ports the tool manages

The browser-facing proxy port defaults to **55142** on plain HTTP and **55143** with HTTPS, and the tool switches between the two when you turn HTTPS on or off. The Viz Mosart server ports on the Studios page default to **55167**, or **55168** when you supply your own certificate.

In both cases the tool only rewrites a port that still holds one of those defaults. A port you typed yourself is never touched, so if you use a custom port, remember to update it yourself after changing HTTPS or the certificate mode. The Web Apps server behind the proxy uses 65142 and 65143 and is not exposed on this page.

Click **Save** after changes.

5.9.2 Serve Config Tool on the web

Serve Config Tool on the web, below the proxy setup method, decides which computers can open the Configuration Tool. Off, the default, the tool answers only in a browser on the Web Apps server itself, and other

Copyright © 2026 Vizrt

computers are refused. On, any computer that can reach the Web Apps server can open it. The change applies as soon as you save.

While it is off and sign-in is off, the tool also answers only on the server's own names: `localhost`, its machine name or address, or the web server and proxy host names set on this page.

> **Note**: Turn it on only together with HTTPS and sign-in. With sign-in off, other computers can use Home but cannot change settings, which takes the Configuration Tool window on the server, run by an administrator. See Configuration Tool and Authentication.

5.9.3 Sign-in and sessions

The sign-in and session controls are rendered on this page in the Configuration Tool, in the **Authentication (OIDC)** block, but they are documented together with the rest of authentication on the Authentication page.

Two of them are new in 3.3 and sit immediately below **Scopes**:

- **Session timeout**: how long an unused session survives before sign-in is required again, in minutes.
- **Allow session timeout**: one switch per web app, deciding whether that app's open pages keep the session alive or let it time out when idle.

5.9.4 Mosart API Access

The **MOSART API ACCESS** block is shown only when the proxy setup method is **Automatic**. Under a manual, external proxy the tool does not offer it, because the key is attached by the bundled proxy.

The **Authorization Key** is the Viz Mosart server's REST API key, configured on the Mosart server under *Remote Controller Service* > *Security*. When the Mosart server runs in API-key mode, the web proxy attaches this key to the calls that need it, so the browser never sees the key itself. It is used for:

- **The Named Overlays Editor**: reading and writing named overlays, including the *Sync to backup server* function.
- **Clip search against older Viz Mosart servers** (before 5.16). From Viz Mosart 5.16, clip search uses the server's live connection and does not need the key.
- **Smart Panel button presses** from paired devices, which the Web Apps server forwards to the Mosart server with this key. See the Panel access section of App Configuration.
- **Availability checks** the Configuration Tool itself runs, for example whether the Mosart server's settings editor is reachable.

> **Note**: The same key is sent to **every** Viz Mosart server the Web Apps talk to: main, backup, and every studio's servers in a multi-studio installation. All of them must therefore be configured with the same REST API key.

> **Info**: With OIDC authentication enabled on the Mosart servers, the REST API key is not accepted there and this field is not used: the apps send the signed-in user's token instead. See the Authentication page.

Copyright © 2026 Vizrt

5.9.5 Media Sequencer

Smart Panel can drive a Media Sequencer directly, for button actions that take an MSE graphic and for asset lists that read their content from a Media Sequencer playlist or show.

Under **MEDIA SEQUENCER**, enter the **Address** of the Media Sequencer and its REST API **Port**. The default port is 8580. A **Pilot Edge Action Panel** can also use this Media Sequencer as its playout target, which needs an address other than `localhost`.

5.9.6 Viz Arc

Smart Panel buttons can execute Viz Arc actions. To make them available, enter the **Address** and **Port** of the Viz Arc REST web service under **VIZ ARC**. The default port is 9004. The Configuration Tool probes the host as you type and reports whether the service answers.

> **Note**: The Viz Arc REST web service must be running on that host, and it requires its own licence. The Web Apps server calls Viz Arc on behalf of the browser, so only the Web Apps host needs network access to it.

5.9.7 Pilot Data Server

Smart Panel's **Pilot Edge Action Panel** widget frames a Viz Pilot Edge template, picked by title from a Viz Pilot Data Server. Enter the server once here; each widget then only carries its template and playout target.

Under **PILOT DATA SERVER**, enter the **Address**, a host name or IP address only, with no `http://` or `https://`, and the **Port**. The Pilot Data Server's default ports are 8177 for HTTP and 7373 for HTTPS: enter the one it serves. An empty **Port** means 8177. HTTP or HTTPS is detected automatically on that port. The Configuration Tool probes the host as you type, reports whether it answers, and refuses to save until it does.

- **Pilot Edge**: which Viz Pilot Edge install Smart Panel frames, offered only when the Pilot Data Server you entered lists more than one. The Configuration Tool keeps a choice the server still lists, or else picks the standard install (*pilotedge*). If the server lists several and none is the standard install, choose one: until you do, Save is refused with *Choose the Pilot Edge to use.*
- **Template Builder**: which Template Builder the widget's **Edit in Template Builder** opens, offered only when the Pilot Data Server lists more than one, and chosen the same way (standard install *templatebuilder*).
- **Pilot sign-in**: how Pilot Edge signs in to a secured Pilot system. *Default (dynamic)* lets Pilot Edge choose: a popup, falling back to a redirect. *Popup* forces a popup with no fallback. The sign-in happens in the operator's browser.
- **Client ID (optional)**: a public OIDC client ID to sign in with, overriding the Pilot system's own default. It is sent whichever sign-in method is chosen.

> **Info**: If you leave **Address** empty, the Web Apps use the server of a Viz Pilot Edge plugin configured for Showmaker or the Named Overlays Editor, and the empty field shows that server, for example *From Pilot plugin: pds01*. The sign-in settings in that plugin's URL then apply too: a popup sign-in set there overrides *Default (dynamic)*, and its client ID is sent unless you enter a **Client ID**. A plugin on `localhost` is not used, because the browser would then look for the Pilot Data Server on the operator's own machine.

Copyright © 2026 Vizrt

Note: Unlike Viz Arc, the Pilot Data Server is reached by the operators' browsers directly, so it must be reachable from every machine that runs Smart Panel, and from the Web Apps host. When the Web Apps use HTTPS, browsers block a Pilot Data Server on plain HTTP: use its HTTPS port (7373 by default).

5.9.8 Video Clips

Video clips (PACKAGE templates) can be presented with various details. This setting controls how clips are labelled in the web apps that list them, such as Smart Panel.

The dropdown is labelled **Video clip is displayed with** and offers five choices:

Option	Shows
Name	The clip name. This is the default.
Description	The clip description only.
Description or Name	The description, falling back to the name for a clip that has no description.
Description and Name	Both, description first.
Name and Description	Both, name first.

5.9.9 TriCaster Live Previews

If your studio uses a TriCaster, Smart Panel can show its video inputs and outputs as *Live previews*. The Mosart web server proxies the streams on behalf of the browser, so it needs the TriCaster address and a user account.

1. In the Configuration Tool, open **Server Configuration** and scroll down to **TRICASTER FOR LIVE PREVIEW**.
2. Enter the TriCaster **Address** and the **Username** / **Password** of an account with access to the TriCaster web browser URL.

After this, TriCaster sources become selectable as preview sources in the relevant Web App.

Info: NDI sources need no configuration here. Mosart Web Apps discovers NDI sources on the network itself, so a Smart Panel live video element can list and preview them directly.

Copyright © 2026 Vizrt

5.10 HTTPS and Certificates

Mosart Web Apps communicates through a proxy server that must be configured. This page covers the proxy setup methods, and how HTTPS and certificates work across the Mosart network.

- Proxy setup method
  - Automatic proxy setup
  - Manual proxy setup
- Certificates
  - Automatic (built-in CA)
  - Custom certificate
  - Which mode to choose
  - Certificate requirements
  - Certificate installation
- Mosart servers over HTTPS

5.10.1 Proxy setup method

You can choose to use the onboard proxy bundled with Viz Mosart or an external proxy. Click either

- **Automatic proxy setup** to use the onboard proxy server (Caddy v2), or
- **Manual proxy setup** to use your own proxy server.

Automatic proxy setup

With the onboard proxy web server, configurations are automatically generated based on the settings displayed on the **Server Configuration** page, and all default settings are pre-filled. Selecting HTTPS automatically configures both the Mosart Web Applications and the Mosart Web Proxy (Caddy). You then choose how certificates are provided, either the built-in certificate authority or your own certificate.

> **Info**: If the Mosart Web Proxy and the Mosart Web Applications are configured with *HTTPS* using your own certificate, you must also configure **HTTPS for the Mosart servers** (main and backup, if applicable). See Mosart servers over HTTPS below.

Manual proxy setup

You can use an external proxy server, but it must be manually configured. If the external proxy server is configured with *HTTPS*, you must also set up HTTPS for the Viz Mosart Web Applications and Mosart servers.

> **Tip**: You can examine the Caddy file generated when *Automatic proxy setup* is selected to understand how to set up an external proxy. This Caddy file resides at C:\ProgramData\Mosart Medialab\Mosart Web Apps.

1. From the **Server Configuration** page, select **Manual proxy setup**.
2. Under section **PROXY WEB SERVER** in field **URL**, enter the URL address to the external proxy server. This must be a valid web URL starting with http or https.
3. Select the **HTTPS** check-box. There is no certificate choice to make here: the **Certificates** control appears only under the automatic proxy setup, because the built-in authority lives inside the bundled proxy and

Copyright © 2026 Vizrt

cannot serve an external one. Choosing HTTPS with a manual proxy therefore always takes the custom-certificate route, and the fields presented are **Web Server Address**, **Certificate file**, **Private Key file** and **Password**.

4. **Web Server Address**: The IP or the hostname of the Viz Mosart web server. The system attempts an auto-detect. If the *FQDN is required* (for example, by a certificate), provide this instead. This field is shown only in the manual setup; under the automatic setup that address is the **PROXY WEB SERVER** field on the **Server Configuration** page.
5. **Certificate file**: The certificate in encrypted PEM format. If FQDN is specified in the certificate file, the Mosart web server address must also be a FQDN.
6. **Private Key file** and **Password**: the matching private key, in encrypted PEM format, and the password that decrypts it.
7. Click **Save**.

5.10.2 Certificates

With HTTPS enabled under the automatic proxy setup, the **Certificates** setting offers two approaches.

Automatic (built-in CA)

The bundled Mosart Web Proxy contains its own certificate authority. Choose **Automatic (built-in CA)** and the proxy mints the certificates it needs itself, the first time it starts. There is no certificate file to obtain, install or renew.

Because the certificates are issued by a private authority, browsers do not trust them until you say so. The tool therefore shows the authority's **Root certificate** as a read-only path on the server. In the Configuration Tool window on the server, **Open folder** beside it reveals the file on disk. Distribute that root certificate to the machines that will use the Web Apps and install it as a trusted root, and the warnings stop.

The root certificate is written to:

`C:\ProgramData\Mosart Medialab\Mosart Web Apps\WebProxyData\pki\authorities\local\root.crt`

> **Note**: The root certificate is only created once the web proxy has started, so save the configuration first. Until then the tool still shows the path, but there is no file there yet: in the Configuration Tool window, Open folder says that the folder does not exist yet.

With the built-in CA, the proxy terminates HTTPS for the browser and talks **plain HTTP to everything behind it**: the Mosart Web Apps server and the Viz Mosart servers, including when those run on other machines. Browser traffic is encrypted; the traffic from the proxy onwards is not.

This is also why the built-in CA is only available with the automatic proxy setup, since it lives inside the bundled proxy and cannot serve a manual, external proxy, and why the Authentication section requires custom certificate mode.

Custom certificate

Choose this to supply your own certificate, which is what you need when your organization issues certificates centrally, when the same certificate must cover several Mosart components, or when you want to enable **Authentication**. Provide the certificate file, the private key file and the key password.

Copyright © 2026 Vizrt

Certificate file (path on the Mosart server) and **Private Key file (path on the Mosart server)** take the path of each file on the machine that runs the Mosart Web Apps server, for example `C:\certs\server.crt`. Type or paste it: the server reads the files there, whichever computer you are using the tool from. In the Configuration Tool window on the server, **Choose file** beside each field opens a file dialog instead. **Password** is an ordinary field and can be cleared. The tool validates all three as you set them and reports the result beside each field, so a wrong password, a key that does not match the certificate, an expired certificate, or a certificate that does not cover the configured host is caught before you save. A certificate covers the host the way a browser judges it: case does not matter, and a wildcard name such as `*.studio.example.com` covers one level of names below it, such as `web01.studio.example.com`, but not `studio.example.com` itself or `a.web01.studio.example.com`. It does not check that the key itself is encrypted, so confirm that before you save.

For the Mosart Web Proxy, the configured private key is used to automatically generate the unencrypted PEM format that Caddy requires, which is saved in `C:\ProgramData\Mosart Medialab\Mosart Web Apps`.

Which mode to choose

Choose	When
Automatic (built-in CA)	You want browser traffic encrypted quickly, with no certificates to obtain or renew. You accept installing the authority's root certificate on each client machine, and that traffic between the proxy and the servers behind it stays unencrypted. User sign-in is not available in this mode.
Custom certificate	You want user sign-in (OIDC), or you want the traffic to the Mosart Web Apps server and the Viz Mosart servers encrypted as well. Also the natural choice when your organization already issues certificates, since client machines then trust them without further steps.

Certificate requirements

The requirements below apply when you supply your own certificate. With the built-in CA, the proxy handles all of this itself and you only have to distribute its root certificate.

- **Type**: X.509 certificates.
- **Format**: PEM-encoded certificates and private keys. For the Mosart Web Proxy (Caddy), the key must be in unencrypted PEM format, which is automatically generated by Mosart from the provided encrypted key file.
- **Encryption**: Private keys must be encrypted and provided with the corresponding password. You can verify that the key is encrypted by opening it in an editor and checking that it starts with "-----BEGIN ENCRYPTED PRIVATE KEY-----".
- **Host coverage**: The certificate must include the host where the Mosart Web Applications are installed. The hosts provided in the Mosart Web Config tool are validated against the certificate the way a browser judges them: case does not matter, and a wildcard name covers one level of names below it. If the hosts in the certificate are the Fully Qualified Domain Name (FQDN), then the hosts provided in the Mosart Web Config Tool must be the FQDN as well. If the same certificate and key files are used across all Mosart components

Copyright © 2026 Vizrt

(Mosart web server, Mosart web proxy, Mosart server main, Mosart server backup), the certificate must specify all relevant hosts or their FQDNs.

Certificate installation

When Mosart Web Applications are configured with HTTPS, all communication between clients, proxy servers, and content servers must be secured: the Mosart web clients (*Rundown Viewer*, *NRCS Plugin*, *Smart Panel*, *Showmaker*), the Mosart web server, the proxy server, and the Mosart server (*Mosart Remote Control Service*).

Certificates must therefore be trusted on any machine where Mosart Web Apps are used. Ensure the certificates are installed on:

- Mosart Web Applications / Mosart Web Proxy.
- Any Mosart server.
- Any machine running Mosart web clients.

If the certificate is not installed on a machine using a Mosart Web App, the browser will warn of an insecure connection, suggesting you proceed at your own risk. However, if the Mosart Web App is integrated as a plugin within other applications (for example, into ENPS), there may be no warning, and instead issues like pages displaying blank may be experienced.

5.10.3 Mosart servers over HTTPS

When the Web Apps run HTTPS with your own certificate, the Mosart servers (main and backup) must serve HTTPS too, so that rundown updates and commands between the Viz Mosart server and the web applications stay secured.

HTTPS on the Mosart server is configured in the Mosart settings under *Remote Controller Service* > *Security*, followed by a restart of the Mosart Remote Control Service on each server. The procedure is described in the *Viz Mosart Administrator Guide*, chapter *Security in Viz Mosart Ecosystem*, section *Securing the Mosart Server*.

The Configuration Tool moves the Mosart server ports for you when you change certificate mode: it writes **55168** when you supply your own certificate and **55167** otherwise, because the Viz Mosart servers serve HTTPS only with a custom certificate. It only retargets a port that still holds one of those two defaults, so a port you typed yourself is left exactly as you set it and has to be changed by hand. Check the **Studios** page if you use bespoke ports, and check each studio's **backup** port as well: a pair that works until a failover is usually a backup left on the HTTP port.

Copyright © 2026 Vizrt

5.11 Authentication

Viz Mosart Web Applications supports authentication using *OpenID Connect*. This page covers the Web Apps half of the setup: what to enter in the Configuration Tool, and the roles the web apps enforce.

> **Info**: This configures the Web Apps half only. For the whole picture, including the identity provider requirements, securing the Mosart server itself, and the end-to-end verification, see the *Viz Mosart Administrator Guide*, chapter *Security in Viz Mosart Ecosystem*.

- Prerequisites
- Roles
  - Showmaker
  - Smart Panel
  - Named Overlays Editor
  - Configuration Tool
- Setup
- Session timeout
  - Before the session ends
- Provider-specific newsroom authentication
  - iNEWS
  - ENPS
  - Octopus

5.11.1 Prerequisites

- HTTPS with a **custom certificate** has to be configured to use authentication; the Authentication section only appears in that mode. See the HTTPS and Certificates page.
- You need a third party OpenID Connect authentication service, such as Keycloak, Okta, or Microsoft Entra ID. (If you are already using Vizrt's asset management system, *Viz One*, you should already have a Vizrt SSO Keycloak authentication server.)

5.11.2 Roles

Roles grant user permissions: what a user can do with the app. Roles are assigned in your authentication provider, not in the Configuration Tool, either to a single user or to all members of one or more groups. Role *names* are predefined and fixed by Vizrt, but you are free to define or reuse groups as you like. Each application reads them from the `roles` claim.

Showmaker

Role name	Permissions
`showmaker-viewer`	View only.

Copyright © 2026 Vizrt

Role name	Permissions
`showmaker-admin`	Full control.

Smart Panel

Role name	Permissions
`smartpanel-viewer`	Open and use panels, but not edit them.
`smartpanel-admin`	Full control, including creating and editing panels.

Named Overlays Editor

Role name	Permissions
`mosart-viewer`	View the overlay list and overlay properties.
`mosart-admin`	Create, edit, delete and import overlays.

⚠ **Note**: The Named Overlays Editor roles are `mosart-admin` and `mosart-viewer`. From Viz Mosart 5.16 with Mosart Web Apps 3.2, these roles are also enforced by the Mosart server on its Named Overlays REST API, so they gate the API as well as the editor UI. Earlier documentation used the names `namedoverlays-admin` and `namedoverlays-viewer`; those names are not read by any current version.

Configuration Tool

Role name	Permissions
`configtool-admin`	Change the configuration. A signed-in user without it can use only **Home** and **Help & Resources**: launching the web apps, seeing whether the services are up, and reading the guides. **Mosart Server settings**, under **Help & Resources**, also needs the role.

With OIDC authentication enabled on the Mosart server, pushing the Showmaker MOS settings to Manus Administrator from **App Configuration** also needs the `mosart-admin` role, which the Mosart server checks. Push them from the Configuration Tool in a browser, not from the Configuration Tool window. Without the role, the

Copyright © 2026 Vizrt

Configuration Tool says **Your account doesn't have RCS's mosart-admin role.** For the other messages it can show, see App Configuration.

Note: Roles are only enforced when authentication is enabled. With authentication *off*, every user has full control. The Configuration Tool is the exception: with authentication off, only the Configuration Tool window on the Web Apps server, run by a Windows administrator, can change settings. Assign roles *before* enabling authentication: in the other apps, a signed-in user with no role gets an unauthorized-access screen, not a read-only view. In the Configuration Tool, they can use only **Home** and **Help & Resources**, apart from **Mosart Server settings**.

Info: The roles must reach the token as a flat, top-level claim named exactly `roles`, present in **both the ID token and the access token**. Keycloak needs a protocol mapper for this; its default nested `realm_access.roles` is not read. See Vizrt SSO (Keycloak) for a walkthrough, and the *Viz Mosart Administrator Guide*, chapter *Security in Viz Mosart Ecosystem*, for the details of which layer reads which token.

5.11.3 Setup

1. Create a new Client/App registration according to your authentication provider guidelines:
    a. Keycloak guide
    b. Keycloak guide for Vizrt SSO
    c. Entra guide
    d. Okta guide
2. Your authentication provider must provide the Issuer URL/Authority URL, the Client ID and the Client Secret.
3. Open the Mosart Web Application Configuration Tool and select **Server Configuration**.
4. Select the **Authentication (OIDC)** check-box and enter the details obtained at step 2. The check-box is described as "OpenID Connect allows this app to provide secure sign-in and verify your identity details." It appears only while the certificate mode is **Custom certificate**, so with the built-in CA there is no authentication section to enable.
5. Optionally add `offline_access` to the **Scopes** field. **Scopes** is a space-separated list of scopes requested *in addition* to `openid`, `profile` and `email`, which are always sent, so to add this one the field contains just `offline_access`.
6. Decide how long a signed-in session lasts, and which apps it applies to, using **Session timeout** and **Allow session timeout** in the same block, below **Scopes**. See *Session timeout* below.
7. Click **Save**. Authentication is now enabled.

Info: Saving after turning **Authentication (OIDC)** on or off restarts the Web Apps server, which the Configuration Tool does by itself, and then reloads the tool so that you sign in with the new settings. Give yourself the `configtool-admin` role first, or the tool shows you only Home once sign-in is on. If a sign-in setting turns out to be wrong and nobody can sign in, open the Configuration Tool window on the Web Apps server as a Windows administrator: it changes settings without signing in.

Copyright © 2026 Vizrt

Info: What `offline_access` is for. It requests a refresh token from your identity provider, which lets the web apps renew the *access token* they present to the Viz Mosart server. Without it, an app that has been open for a long time eventually cannot reach the Mosart server any more. It has nothing to do with how long you stay signed in to the web apps themselves: that is the **Session timeout** setting, and renewing a Web Apps session reissues the existing sign-in locally without contacting the identity provider.

Info: On a first-run installation the same settings are collected by the **Quick setup** wizard. Choose **Secure with HTTPS** on the **Access** step, then **Custom certificate** on the **Security** step, and a **Require sign-in (OpenID Connect)** toggle appears with **Issuer URL**, **Client ID**, **Secret** and **Scopes (optional)**. The review step warns **Sign-in won't be enforced** if a configuration arrives with OIDC on but no custom certificate, which is the state to watch for when carrying settings over from another machine.

Note: Enabling OIDC authentication on the **Mosart server** (a separate setting, on the server) switches off its REST API key, which affects integrations that call the Mosart REST API directly. Read *Choosing an Authentication Mode* in the *Viz Mosart Administrator Guide*, chapter *Security in Viz Mosart Ecosystem*, before enabling it in production.

5.11.4 Session timeout

With authentication enabled, a signed-in session ends after a period without use and the user has to sign in again. Both how long that takes, and which apps it applies to, are configurable. Both controls are in **Server Configuration**, in the block that appears below **Scopes** once **Authentication (OIDC)** is selected.

Session timeout is how long an unused session survives, in minutes. The default is **30**, and the permitted range is 5 minutes to 10080 minutes, that is one week. A value outside the range is clamped to the nearest limit, and one that is not a whole number falls back to 30; either case is written to the server log at startup.

Allow session timeout carries one switch per web app, with a line above it counting how many are switched on, for example *3 of 7 apps can time out when idle.* A switched-on app reads **Renews while in use; can time out when idle**. A switched-off app reads **Stays signed in while open**, and keeps its session alive for as long as its page is open, whether or not anyone is at it.

The defaults are chosen so that nothing on air is signed out mid-show:

Can time out when idle	Stays signed in while open
Showmaker Template Editor Named Overlays Editor	Smart Panel Prompter Rundown Viewer NRCS Plugin

Note: An app that is not listed in the configuration at all defaults to timing out. That is the safe direction for an app added in a future release, but it means a hand-edited configuration file can leave a surface you expected to stay signed in timing out instead. Check the switches rather than the file.

Copyright © 2026 Vizrt

Before the session ends

The window is an idle window: it runs from the user's last activity, not from when the page was loaded or last renewed. Typing, clicking, scrolling, the scroll wheel, moving the mouse, touch and any text input all count.

Where an app embeds a panel from another product, such as Viz Pilot Edge, that panel's own events cannot be seen from the host page. Having the focus in it therefore counts as activity for up to five minutes, and anything the panel sends back, such as an item dropped into a story, counts in full and starts the window again.

Ninety seconds before the session would end, an app that can time out shows a dialog headed **You will be logged out soon**, reading "For your security, we log you out automatically when you have not been active for a certain period of time.", with a live countdown that turns urgent for the last thirty seconds.

- **Continue session** buys a fresh full window and returns the user to what they were doing. Nothing reloads and nothing is lost.
- **Log out now** ends the session immediately.

If the countdown runs out, the app shows **Session timeout!** with a **Login** button.

Info: The countdown comes from the server rather than the browser's own clock, because a machine that has been asleep, or a tab the browser has throttled in the background, cannot be trusted to have counted correctly. This is also why a laptop reopened after a long sleep goes straight to the expired dialog rather than showing a warning first.

Note: Renewal reissues the existing sign-in without contacting the identity provider. The session timeout therefore also bounds how long a disabled account, or a changed set of roles, keeps working in the web apps. Shorten it if your organisation expects a revoked account to lose access promptly.

5.11.5 Provider-specific newsroom authentication

iNEWS

Authentication will only work if you are using the NRCS plugin with the NRCS plugin *iNEWS Client Addon*. Authentication will not work by hosting the Plugin as a regular HTML plugin.

ENPS

To use the NRCS plugin with ENPS, first add your ENPS server machine to the list of allowed frame-ancestors in the Content Security Policy on your Authentication Provider.

Typically this is a simple edit of the Content Security Policy, from

```
frame-src 'self'; frame-ancestors 'self'; object-src 'none';
```

to

```
frame-src 'self'; frame-ancestors 'self' {Your ENPS server}; object-src 'none';
```

Copyright © 2026 Vizrt

- **Keycloak**: Go to **Realm Settings > Security Defenses**.
- **Okta**: Follow this Okta guide.
- **Entra**: Since you cannot edit the Content Security Policy for Microsoft Entra ID, authentication with Entra ID for the NRCS plugin in ENPS is not supported.

Octopus

NRCS plugin authentication is not currently supported with Octopus.

Copyright © 2026 Vizrt

6 NRCS Plugin

The NRCS Plugin is the tool journalists use to add Viz Mosart template instructions to a story. Find the template you need, adjust its production details, then place it in the story either by dragging it in or with a single **Add to Story** action.

The plugin runs in a browser and is hosted inside the system you already work in: your newsroom computer system (ENPS, iNEWS, Octopus or OpenMedia), *Viz Pilot Edge*, or the Mosart web apps *Showmaker*, *Template Editor* and *Smart Panel*. Some controls adapt to the host, so a few of the actions described in this section are deliberately absent in some of them.

- Working with the NRCS Plugin
- NRCS Plugin in Pilot Edge
- Setup and Administration

Info: For news-breaking details on the **Mosart NRCS Plugin**, please refer to the *Mosart Web Apps Release Notes* for your version of Viz Mosart at the Vizrt Documentation Center.

Copyright © 2026 Vizrt

6.1 Working with the NRCS Plugin

The NRCS Plugin delivers the functionality previously provided by the legacy *Viz Mosart ActiveX* Plugin.

- User Operations
- Keyboard Shortcuts
- Getting Help
- Error Handling
- Session and Sign-in
- API Documentation

6.1.1 User Operations

Searching for a Template

Choosing a gallery and template set

The header shows the gallery and template set the plugin is currently working with, the gallery name above and the template set name below it. Click either to open the **Select Template Set** dialog.

- **FILTER BY SET NAME** narrows the list of template sets as you type.
- **CHOOSE GALLERY** lists the available galleries. This list only appears when there is more than one gallery to choose from, so an installation with a single gallery never shows it.
- Click a template set to apply it and close the dialog. If nothing matches your filter, the dialog reports *No template sets found.*

Filtering by template type

The **Template Types** list on the left filters the template listing to a single type. **ALL** is always the first entry and shows everything. Each type carries the same colour and icon that Viz Mosart uses for it, and it is named with your newsroom's own alias for that type rather than a fixed Vizrt name.

> **Info**: Your administrator decides which types appear and in what order, so this list can differ between installations. See App Configuration.

Searching

Type in the **Search** box to narrow the listing. The list filters live as you type, highlighting the matching text, and matches against both the template *variant* name and the *template type* name. Several words act as an AND: every word you type must match one of the two, in any order. Clear the box with the **x** to return to the full listing.

> **Info**: Search works within the type you have selected in **Template Types**. If a template you expect is missing, select **ALL** and search again.

Copyright © 2026 Vizrt

Tip: **Ctrl+F** puts the cursor straight into the **Search** box from anywhere in the plugin. Press **Enter** in the box to move focus to the selected template, then use the arrow keys to move around the listing.

Modifying a Template

Click a template in the listing and its production details appear in the editor on the right. The editor shows only the sections that apply to the template you selected, and each section has its own **RESET** button that restores that section to the values held in the template itself.

Tip: Hover a **RESET** button to see exactly what it will restore, for example *Reset to default: MIX 12*. Pressing **R** resets every section currently shown in the editor.

To modify production characteristics

Newsroom Variables

- Click on a template in the search listing. Properties (called *Variables*) that have been added to the template (using Viz Mosart's Template Editor) appear to the right.

Note: *Variables* are called *Newsroom tags* in the Viz Mosart Template Editor.

- You can temporarily override the value of any variable, for this story item only. The template itself is not changed. For example, in a *Video Effects* template, you may wish to change the source of the second input.

Note: The set of variables comes from the template. You can change their values here, but you cannot add a variable that the template does not define. To add one, edit the template in the Viz Mosart Template Editor.

Each variable is presented with the control that suits it:

Variable	Control
A variable bound to a Viz Mosart list: mixer inputs, router sources, router destinations, engine presets or engine top scenes	A drop-down of the values read from your Viz Mosart configuration
A true/false variable	A check box
A timecode variable	A time field in *mm:ss* or *hh:mm:ss*, with step buttons. A duration variable is labelled **Duration**.
The video clip path variable	A text field with a search button that opens the clip picker. See *Selecting a video clip* below.

Copyright © 2026 Vizrt

Variable	Control
Anything else	A text field

Transitions

- The **Transition** field initially displays the *default* transition type. This value indicates the most significant template behavior.
- You can temporarily *modify* this transition type, and then later restore the default transition (as originally set in the Template Editor).
- The three choices are **MIX**, **WIPE** and **EFFECT**. MIX and WIPE take a duration in **Frames**; negative values are rejected. EFFECT takes a named effect from a drop-down of the effects configured for the gallery.
- Switching to a different transition type sets the rate to 0. Switching back to the template's own type restores that template's rate.

Note: You can restore the default transition by clicking the **Reset** button.

Timings

Templates that can be timed, that is lower thirds, accessories, play sound and commands, also offer a **Timings** section:

- **AUTO** takes the item automatically, and offers both an **In** and an **Out** time.
- **MANUAL** waits for the operator to take the item, and offers only an **Out** time.

Commands and direct takes

A command template shows a **data** field carrying the command's payload. The one exception is the *Direct Take* command, which instead offers a searchable drop-down of the direct takes configured in Viz Mosart, listed by name, so you pick a direct take by name rather than typing its recall number.

Selecting a video clip

For a template with a video clip path, click the search button in the clip field to open the **Select video clip** dialog. Type in **Enter clip name** and the plugin asks the Viz Mosart media service for matching clips.

- Results are listed by their **full clip name**. Where a name is too long for the row, hovering it rolls the text sideways so you can read the end of it.
- An exact match for what you typed is listed first; the rest keep the media server's own order.
- Long result sets are shown 50 rows at a time, with a **Show more** button that reports how many remain. A line above the list tells you how many of the total you are seeing.
- Where the media server provides them, each row shows a **thumbnail** that can be expanded to a preview, and an information icon carrying the clip description.
- Opening the dialog with an empty search box lists the clips you used most recently, on this machine, so common picks need no search at all.
- Select a clip by clicking it, or keep your hands on the keyboard: **Enter** highlights the first result, the **arrow keys** move the highlight, and **Enter** again takes the highlighted clip and closes the dialog. Click **OK** to confirm, or **Cancel** to leave the field as it was.

If a search cannot be answered, the dialog says which part failed, so you know whether to look at the clip name or call your administrator:

Copyright © 2026 Vizrt

Message	Meaning
No clips found matching your search. Try again or check for typos.	The media server answered, and nothing matched.
Couldn't reach Media Admin	The Mosart server accepted the search but the media service did not answer in time, usually because Media Administrator is stopped or still starting.
Couldn't reach the Mosart server	The Mosart server or the Remote Control Service could not be reached at all.

Adding Template Details to the NRCS

- Drag and drop the template into the required location of the story slug. Drag the template's row in the list, or, with a template open, its name at the top of the editor (not while editing an item opened from the story).
- In a MOS-based NRCS, in addition to drag and drop, the **Add to Story** button can be used to add a template to a story.
- There is an **Add to Story** button in two places: on the template row itself, which appears when you hover the row, and below the editor.
- Two shortcuts do the same thing: pressing **Enter**, and double-clicking the template row.
- An administrator can turn the **Add to Story** buttons off, with **Show Add to Story buttons** in the Configuration Tool's App Configuration. Templates are then added by drag and drop only: the buttons are hidden, **Enter** and double-click do not add a template, and a drag handle appears on the row under the pointer. Items opened from the story can still be updated or replaced. In iNEWS, its own **OK** and **Apply** buttons still add a template.

Note: In iNEWS there is no **Add to Story** button. Use iNEWS's own **OK** and **Apply** buttons instead, as described under *Modifying Existing MOS Objects* below. In Viz Pilot Edge, drag and drop and **Add to Story** are both absent, because Pilot Edge provides its own controls; see NRCS Plugin in Pilot Edge.

Info: When the plugin is opened from the Showmaker **Show Assets** page, the button reads **Add to show assets** and saves the item to the show's assets instead of into a story.

Info: The names of both the *gallery* and the *template set* of your selected template are always included with the MOS object that you store as a story item in your NRCS rundown.
Any future use of this NRCS MOS item will always point to exactly the same template (same *gallery*, *template set* and *template name*) when it displays in the Mosart NRCS Plugin.

Refreshing Template Changes

If changes are being made to the active template set (by another operator using Viz Mosart's Template Editor), click the **Refresh** button from **Settings** to get the latest template changes.

Copyright © 2026 Vizrt

You do not have to watch for this. When the templates change in Viz Mosart and the change affects what the plugin shows (the template sets of the gallery, its direct takes, or the templates in the set on screen), a message appears that reads *Templates were changed in Viz Mosart.* Click **Refresh** in the message to load the changes, or close it to keep working with what is on screen. The message stays until the templates are reloaded or you close it, and appears again on the next change. Changes to the templates of other template sets, or to other galleries, do not show it, unless you closed the message earlier without loading the changes: then the next change to any template shows it again.

Note: The message only appears when the Viz Mosart templates are read from a folder, with **Location** set to **Local** in App Configuration. When the templates are stored in a database, no message appears, so refresh to see changes.

Any refresh also loads the changes and closes the message: **Shift+R**, the **Refresh** entry in the **Settings** menu, confirming **Plugin Configuration**, and opening an existing story item. A short confirmation appears once the reload is done. If the reload fails, the message comes back so that you can try again.

A refresh keeps your work on the selected template. A field you have filled in or changed keeps your value, and a field you have not touched takes the template's new default. Fields added to the template appear, and fields removed from it go. While you are editing an existing story item, all of its values are kept.

Optimizing Your NRCS Plugin Workspace

Collapsing the Type listing

You can save some screen space taken by the Mosart NRCS Plugin by collapsing the **Template Types** panel:

- Click the **Template Types** button at the top of the panel, whose tooltip reads *Hide labels*. Collapsed, the list keeps each type's colour and icon and drops only the labels, so you can still filter. Click it again, when the tooltip reads *Show labels*, to bring the labels back.
- You can customize the number of columns to adjust the template list size. Selecting the last option will automatically fit the number of columns to the window size.

Info: In a narrow window the plugin looks after this itself: the **Template Types** panel collapses on its own, and the **Settings** button moves from the top bar to the bottom of the left-hand rail. The divider between the template listing and the editor can also be dragged, so you can give either side more room.

Scaling the View

As the Plugin is a web application, you can set your preferred scale by *zooming in and out*.

- For most browsers, you can use the shortcut **Ctrl+mouse wheel**.

Copying the MOS Object Details

You can work with the template's raw MOS data by first copying it to your PC's clipboard.

- Hover the template card at the top of the editor and click the **Copy** icon that appears on it. A short confirmation tells you the MOS object is on the clipboard.

Copyright © 2026 Vizrt

Note: This action is not offered when the plugin is hosted in Viz Pilot Edge.

Modifying Existing MOS Objects

Your rundown will usually contain slugs with MOS objects that have been created with either this Mosart NRCS Plugin or the legacy component, Viz Mosart ActiveX.

To modify an existing MOS object

1. On the selected rundown story, double-click the slug or grommet that holds the story's MOS object.
2. The corresponding Viz Mosart template, along with control details such as *newsroom tags*, *crosspoints*, *transitions*, and *timings*, is displayed in the Viz Mosart NRCS Plugin. Make the modifications directly in the Mosart NRCS Plugin window.
3. When finished editing, apply the changes. In the iNews newsroom system, use the **OK**, **Apply**, or **Cancel** buttons (specific to iNews) to modify and close, modify without closing, or cancel the editing, respectively. In a MOS-based NRCS, the plugin includes its own buttons to modify or cancel the editing.

The editing band

While you are editing an existing story item, the editor is framed and carries a band reading **EDITING MODE / Update item**, so it is always clear that what you change will be written back to the rundown rather than added as something new. The information icon in the band explains the same thing. The send button changes with it:

- **UPDATE IN STORY** applies your changes to the item you opened.
- **REPLACE IN STORY** appears when you select a *different* template while editing. The band then reads **EDITING MODE / Replace item** and the editor shows the original template, an arrow, and the template that will replace it, so you can see the substitution before you commit it.

To leave editing without applying anything, use **CANCEL**, the **x** in the band, or press **Escape**.

Info: When the plugin is hosted in Showmaker, moving on to something else closes the pending edit for you, so a later **Update** cannot be applied to the wrong item.

Info: You cannot modify a MOS object that refers to a not-found Viz Mosart template.

To modify MOS objects created with the Mosart NRCS Plugin

- MOS objects created by the Mosart NRCS Plugin include references to the *gallery* name and *template set* name.

Info: The active *Gallery* and/or *Template set* name may automatically change if your selected MOS object references a different gallery/template set. You are always notified if this happens.

- If your rundown item refer to a template set that cannot be found in the gallery, the Mosart NRCS Plugin checks the active gallery and template set for a corresponding template. This valid reference is then applied to the MOS object, when you click **UPDATE IN STORY** (in MOS-based NRCS) or **OK/Apply** (in iNews) on the replacement template.

Copyright © 2026 Vizrt

- Alternatively, you can manually search gallery/template sets to locate a similar template.
  After editing and applying the changes, the new (active) gallery and template set name are stored in your rundown MOS object.

To edit MOS objects created by ActiveX

When working with MOS objects that were created with Viz Mosart's legacy NRCS Plugin (Mosart Active-X) the currently selected gallery name and template set name will always be used as template source.
In the Plugin, ensure to select correct gallery and template set before you start editing.

6.1.2 Keyboard Shortcuts

Open the shortcut list at any time with **K**, or from **Settings > Shortcuts**.

Action	Shortcut
Navigate the template listing	Arrow keys, or **W** / **S** / **A** / **D**
Add to story	**Enter** (with the **Add to Story** buttons turned off, only to update or replace an item opened from the story)
Cancel editing	**Escape**
Reset editor values	**R**
Re-fetch templates	**Shift+R**
Focus the search box	**Ctrl+F**
Show keyboard shortcuts	**K**

Info: The navigation keys are ignored while you are typing in a field, so **W**, **A**, **S** and **D** can still be typed into the search box and into variable values.

6.1.3 Getting Help

The NRCS Plugin documentation is continually updated and available online. This includes both this User Guide and the Release Notes.
You can also send your feedback directly to the Mosart NRCS Plugin design team.

- Click the **Settings** icon at the top right. The menu holds:

Copyright © 2026 Vizrt

Entry	What it does
Refresh	Reloads galleries, template sets and templates from Viz Mosart.
Settings	Opens **Plugin Configuration**: the newsroom system, the **Mos ID**, and a **Responsive layout** option that lets the layout follow the window size.
Shortcuts	The keyboard shortcut list.
Documentation	This user guide.
What's new	The release notes for this version.
Give feedback	Opens a mail to the Mosart team.
About	Version information for the plugin.

Info: The newsroom system is usually detected or supplied by the host, in which case it is not offered for editing in **Plugin Configuration**. **Mos ID** is not shown for iNEWS, which does not use it.

6.1.4 Error Handling

While navigating through templates, if there are exceptional conditions, the Mosart NRCS Plugin displays an error message.

Error messages include the address of the plugin host, which is useful when reporting a problem. Confirmations and warnings clear themselves after a few seconds, while errors stay until you dismiss them.

Please refer to the Troubleshooting or contact Vizrt Support if you encounter any error messages.

6.1.5 Session and Sign-in

This applies only where authentication (OIDC) is enabled for Mosart Web Apps. With authentication off, the plugin never asks you to sign in.

Docked in your newsroom system, or hosted in Viz Pilot Edge, the plugin keeps its own sign-in and holds it for as long as its page is open. It is not signed out because nobody has touched it for a while, so a plugin left docked through a shift stays usable.

Hosted inside Showmaker, the Template Editor or a Smart Panel, the plugin does not manage the session at all. The app around it does, and any countdown or sign-in prompt appears there rather than in the plugin panel. Showmaker and the Template Editor can both time out when idle, so a plugin panel that has gone quiet inside one of them may be waiting on that app rather than on itself.

Copyright © 2026 Vizrt

Info: Working inside the plugin counts as activity for the app hosting it, even though the plugin is a separate frame. Building a graphic in an embedded panel therefore keeps the host app's session alive, and anything you send back into a story renews it in full. See Authentication.

6.1.6 API Documentation

The Mosart REST API is installed alongside the NRCS Plugin. Its endpoints are documented as an OpenAPI (Swagger) page, served at:
http://machine-name:55142/mosart/swagger/index.html

The quickest way to reach it is **Help & Resources > API Docs** in the Configuration Tool, which opens the same page for the host and port your installation actually uses.

Info: Replace *machine-name* and the port with the address of your Mosart Web Apps host. *55142* is the default, and your administrator may have changed it. Over HTTPS the default is *55143*.

Examples

- Retrieve *all* galleries:
  http://localhost:55142/api/v1/galleries
  (A list of gallery names is obtained)
- Choose a gallery name from the list obtained in the previous example. For the sake of this example, assume that *Local* was on the list. (If not, choose some other gallery name.)
  To retrieve the template sets for the gallery named *Local*:
  http://localhost:55142/api/v1/galleries/Local/templatesets.

Copyright © 2026 Vizrt

6.2 NRCS Plugin in Pilot Edge

If you are using the Pilot Edge version 3.0 or later, you can host the NRCS Plugin within the Pilot Edge Plugin.

6.2.1 Hosting NRCS Plugin within Pilot Edge

- Set Up
- Working with the Mosart NRCS Plugin in Pilot Edge
- What is different in Pilot Edge

Set Up

To activate the NRCS Plugin in Pilot Edge

1. Open your Pilot Data Server.
2. In the menu bar, click **Settings**.
3. On **Pilot Settings** at the top right, right-click and select **Add New Optional Parameter**.
4. In the **Name** field, add the name *moseditor_mosart_url*. This now appears under the **Name** column.
5. Under the **Value** column, enter the URL to your Mosart NRCS Plugin.

Tip: You can fetch this address from the Configuration Tool by selecting **App Configuration > NRCS Plugin** and copying the value of **Preview URL**.

6. At the top left, click **Save**.

Note: If you are using the Pilot Edge on **HTTPS**, the Mosart NRCS plugin must also be on **HTTPS**.

Info: Enter the plugin URL exactly as the Configuration Tool gives it. Pilot Edge adds what it needs to identify itself when it opens the plugin, and the plugin recognises Pilot Edge from that, so there is nothing further to append.

Working with the Mosart NRCS Plugin in Pilot Edge

1. In your newsroom system, open the Pilot Edge Plugin.
   A new tab **Mosart** appears at the top**.**
2. Click the tab to open the Mosart NRCS Plugin.
3. **To add a Mosart item to your rundown**, select the item and drag in the **Cross** icon at the top right of the plugin. For a MOS-based NRCS, the **Send to newsroom system** option is also available.
4. **To edit an added item**, Pilot Edge automatically opens the NRCS Plugin inside Pilot Edge. In the iNews newsroom system, use the **OK**, **Apply**, or **Cancel** buttons (specific to iNews) to modify and close, modify without closing, or cancel the editing, respectively. In a MOS-based NRCS, the Pilot Edge plugin includes the **Update in newsroom system** button.

Copyright © 2026 Vizrt

What is different in Pilot Edge

Pilot Edge supplies the controls that place an item in the rundown, so the plugin's own equivalents are deliberately not offered here. Compared with running the plugin directly in a newsroom system:

- There is no **Add to Story** button, and dragging a template out of the plugin does nothing. Use Pilot Edge's **Cross** icon or **Send to newsroom system** instead, and its **Update in newsroom system** button when editing.
- **Copy MOS object to clipboard** is not offered.
- The newsroom system is not shown for editing in the plugin's **Settings**, because Pilot Edge has already told the plugin what it is.

You still select a gallery and template set, filter and search templates, and edit variables, transitions and timings exactly as described in Working with the NRCS Plugin.

Info: Your edits are handed to Pilot Edge as you make them. Every time you select a different template or change a value, the plugin passes the updated Mosart item and its description straight to Pilot Edge, so what Pilot Edge shows for the item always matches what you have in front of you. There is no separate step in the plugin to confirm a change; you commit it with Pilot Edge's own button.

Copyright © 2026 Vizrt

6.3 Setup and Administration

This section describes essential post-installation configurations, usually performed by a system administrator.

- Setting up the NRCS Plugin
- Optional Server Setups
    - Support for Main/Backup Connection
    - Support for HTTPS
- Session Timeout
- Verifying the Mosart Installation
- Template Type Filter Order
- Clip Search
- Hide Effects from Users
- Integrating with an NRCS
    - ENPS
    - iNEWS
    - Octopus
    - OpenMedia
    - Showmaker
- Troubleshooting

6.3.1 Setting up the NRCS Plugin

There is a standard setup that must first be followed. Please refer to the configuration procedure in section App Configuration.

6.3.2 Optional Server Setups

Support for Main/Backup Connection

- If you have a *backup server* for redundancy, you will need to define the environment on the Studios page, where each studio carries its own main server and optional backup.

Support for HTTPS

- If you are using *HTTPS*, you will need define the environment in the menu HTTPS and Certificates.

> **Info**: Make sure to execute *all* instructions, including the one-time edits to the Mosart Remote Control Service. These are only required when you supply your own certificate. If you use the built-in certificate authority, the bundled web proxy handles HTTPS itself and there is nothing to edit. See HTTPS and Certificates.

Copyright © 2026 Vizrt

6.3.3 Session Timeout

Where authentication (OIDC) is enabled, you decide how long a sign-in survives and which apps hold it open. Both controls are on the **Server Configuration** page of the Configuration Tool, inside the **Authentication (OIDC)** block, under **Allow session timeout**. That block appears only when you supply your own certificate. See Authentication.

- Every app has its own switch, and a summary line above them reports how many of the seven can time out. Switched on, an app reads **Renews while in use; can time out when idle**. Switched off, it reads **Stays signed in while open**.
- The **NRCS Plugin** ships with its switch *off*, so a docked plugin keeps its sign-in for as long as its page is open.
- **Session timeout** sets the window in minutes, 30 by default, and applies only to the apps whose switch is on.

Note: Leave the NRCS Plugin on **Stays signed in while open**. A plugin docked in a newsroom system is often left untouched for long stretches while the journalist works in the story text around it, so an idle timeout would sign it out mid-shift and the next drag of a template would fail. When the plugin is embedded in Showmaker, the Template Editor or a Smart Panel, the host app owns the session instead, so those apps' own switches apply.

6.3.4 Verifying the Mosart Installation

1. After following the mandatory App Configuration and Server Configuration procedures, the Windows services that serve the web apps are running. These are *Mosart Web Applications*, which hosts the NRCS Plugin and the REST API it uses, and *Mosart Web Proxy*, which publishes them to browsers. On the Viz Mosart server itself, the *Mosart Remote Control Service* provides the connection to Viz Mosart.
2. You can check that the NRCS Plugin is accessible by either clicking **Home > NRCS Plugin** in the Configuration Tool, or following the **Preview URL** next to the **Copy** button on the NRCS Plugin page of **App Configuration**.

Info: The plugin URL is built from the address of your Mosart Web Apps host, the NRCS type and, where the newsroom system uses one, the *Mos ID*:
http://machine-name:55142/plugin?nrcs={type}&mosid={mosid}

- *55142* is the default port of the Mosart Web Proxy, and *55143* is the default over HTTPS. Your administrator can change both, so always take the URL from the Configuration Tool rather than assembling it by hand.
- *mosid* is omitted for iNEWS, which does not use it.
- *nrcs* is omitted when the NRCS type is set to **Other**. The operator then chooses the newsroom system in the plugin's own **Settings**.

3. To complete the integration, configure the NRCS side, according to the appropriate NRCS as described in section Integrating with an NRCS below (where you will use the copied URL).

6.3.5 Template Type Filter Order

The plugin's **Template Types** list is admin-controlled: you decide the order in which the template type filters appear, once, for everyone using the plugin. Put the types your journalists reach for most at the top. Order is all this setting controls, so no type can be left out here: the plugin shows a filter for every template type your newsroom configuration defines that also has templates in the selected template set.

Copyright © 2026 Vizrt

This is set on the NRCS Plugin page of **App Configuration**. For the procedure, see App Configuration.

> **Info**: The **ALL** filter is always first and cannot be moved or removed. Any supported type you do not place explicitly keeps its original position, after the ones you did place. Where your newsroom defines its own alias names for template types, the rows are shown with those names, so the list reads the same as the plugin does.

6.3.6 Clip Search

Templates that carry a video clip offer a clip picker that searches your media server by clip name. For it to return results:

- **Media Administrator must be running** on the Viz Mosart server. It is the service that performs the search. If it is stopped or still starting, the plugin reports that it could not reach Media Admin.
- The **Mosart Remote Control Service** must be reachable from the Mosart Web Apps host, as for every other Viz Mosart connection.

Authorization key

Current Viz Mosart servers answer clip searches over the Mosart server's own real-time connection, which needs no separate key. Older Viz Mosart versions do not offer that, and the plugin falls back to the REST media search, which does require the **Authorization Key** to be configured. If clip search fails only on an older Mosart server, this is the first thing to check. See Server Configuration.

Multi-studio sites

On a site with several studios, the plugin searches the media of the studio it was launched for, so an operator working on studio B does not get studio A's clips. The plugin URL that the Configuration Tool gives you is not studio-scoped, so append &studioId= followed by the studio's slug to the plugin URL you configure in the newsroom system. Without it the plugin searches the media of the main Viz Mosart server. See Studios.

Clip thumbnails

Where your media server supplies them, clip results carry thumbnails and a preview. TriCaster thumbnails are fetched by the Mosart web server on the browser's behalf, using the TriCaster credentials from **Server Configuration**, so operators do not need network access to the TriCaster and the thumbnails work over HTTPS. If clip thumbnails are blank for a TriCaster source, check the TriCaster address, username and password. See Server Configuration.

6.3.7 Hide Effects from Users

In some production situations, there may be an effect recalled from specific templates, that should only be run under specific circumstances (for example, on an ME row).
Unintentionally selecting such an effect from the NRCS plugin (or the Viz Mosart GUI) can give unplanned on-air consequences.

Copyright © 2026 Vizrt

To hide an effect from the user

Info: This feature requires Mosart Web Apps version 2.5 or later.

1. On the Mosart server open AV Automation.
2. Navigate to **A/V Setup > Effects.**
3. Check **Hide From User** to prevent the effect displaying in the NRCS Plugin (and the Viz Mosart GUI **Transition** panel).
4. Save.
5. Refresh the NRCS Plugin.

Info: **Hide From User** is applied the same way to switcher sources, to individual templates and to whole template sets, not only to effects, so you can keep any of them out of the operator's reach. The deliberate exceptions are the two tools you build with: when the plugin is embedded in the Template Editor, or in the template picker Smart Panel offers in Edit mode, hidden items are shown, because those are the tools used to build the templates and panels that reference them.

6.3.8 Integrating with an NRCS

- The server-based plugin configuration described below avoids having to configure each individual newsroom control system (NRCS) client.
- This section assumes the Mosart NRCS Plugin has been successfully installed and set up, as described in section Configuration Tool.

The NRCS Plugin supports connections to

- ENPS
- iNEWS
- Octopus
- OpenMedia
- Showmaker.

It can also be hosted in Viz Pilot Edge, and inside the Mosart web apps *Template Editor* and *Smart Panel*.

You connect each client NRCS machine to the Viz Mosart server (which hosts the Mosart REST API.)

ENPS

1. In the Configuration Tool select **App Configuration > NRCS Plugin** and copy the value of **Preview URL**.
2. On the ENPS machine, navigate to **System Maintenance > MOS Configuration**.
3. Paste the NRCS Plugin's copied URL.

Info: The URL comprises the Plugin's URL plus Viz Mosart's *MosId* value, in the form: *http://machine-name:55142/plugin?nrcs=ENPS&mosid=xxxx*

- If you need to derive current *MosId* value, see App Configuration.

Copyright © 2026 Vizrt

iNEWS

There are two possible setup routes, depending on whether your iNEWS version supports HTML5 plugins. Please consult the documentation of your iNEWS system to determine whether this is the case.

iNEWS Clients with HTML5 Support

1. In the iNEWS client, from the directory navigation tree, open the file **SYSTEM.MOS-MAP** and verify that between `TABLE-START DeviceTable` and `TABLE-END` there is a line `mosart ...`
   If not, add the mosart reference,
   For example: `mosart MosartNRCSPlugin`
2. For this step and the next, please refer to Avid's MediaCentral Newsroom Management Setup and Configuration Guide, section *Configuring Newsroom Management for HTML5* (chapter *HTML5 Plugins*) .
   If you have at least one HTML5 plugin already defined, you can *skip* this step.
   From Avid's *Configuring Newsroom Management for HTML5*, follow their setup steps **1** to **6**. Please note that at:
   - Step **3**: (name of the new queue): Use the name *HTML-PLUGINS*, as recommended by Avid.
3. Continue with steps **7** to **10**, observing these clarifications:
   - Step **7**: In case you skipped the previous step, *the queue* refers to SYSTEM.HTML-PLUGINS
   - Step **9**: (slug): Enter something like *Viz Mosart NRCS Plugin*. (This is the string that will be shown in the **Plugins** sub-menu of the **Tools** menu.)
   - Step **10**: (parameters): In the **Story** panel, enter 'URL = ', and at that exact spot paste the URL copied from the Configuration Tool (**App Configuration > NRCS Plugin**, the value of **Preview URL**). Then append
     mosItemBrowserProgID = Mosart.ActiveX
     mosItemEditorProgID = Mosart.ActiveX

     Example:
     URL = http://machine-name:55142/plugin?nrcs=iNews
     mosItemBrowserProgID = Mosart.ActiveX
     mosItemEditorProgID = Mosart.ActiveX

Info:
- The parameter *mosItemPlayerProgID* mentioned in the Avid document is not mandatory in this context.
- The URL element *mosid* is not required for iNEWS.

Note: If the old Mosart ActiveX opens instead of the Mosart NRCS Plugin when trying to modify/edit a Mosart item (by double clicking on the item), you must uninstall the Mosart ActiveX.

iNEWS Clients without HTML5 Support

Installing the iNEWS Client Addon

Copyright © 2026 Vizrt

For older versions of iNews without HTML5 support, you need to install a small wrapper utility, the **Mosart NRCS Plugin - iNEWS Client Addon** on each iNEWS client machine.

1. From the Vizrt FTP /products/VizMosart/Latest Version/WebApplications/ download and run the iNEWS Client Addon utility, *MosartNRCSPlugin-iNEWSClientAddon-<version>.exe*
2. From the Configuration Tool select **App Configuration > NRCS Plugin** and copy the value of **Preview URL**.
3. In **Mosart NRCS Plugin - iNEWS Client Addon**, paste-in the URL of the Plugin.

Note: If the old Mosart ActiveX opens instead of the Mosart NRCS Plugin when modifying a Mosart item (by double clicking on it), you must first uninstall the Mosart ActiveX, and then reinstall the *MosartNRCSPlugin-iNEWSClientAddon*.

Info: To change the configured URL after the installer was closed, simply open a command prompt and execute:
"%programfiles(x86)%\Vizrt\Mosart NRCS Plugin - iNEWS\MosartNCSPlugin.exe" url "https://machine-name:55142/plugin/?nrcs=iNews"

Note: If you use authentication, iNEWS only works through this *iNEWS Client Addon*. Hosting the plugin as a plain HTML plugin does not support authentication. See Authentication.

Octopus

1. From the Configuration Tool select **App Configuration > NRCS Plugin** and copy the value of **Preview URL**.
2. Log-in to Octopus with Admin privileges.
3. From the sidebar, navigate to **MOS Devices > Devices**, select *mosart* and click **Edit**.
4. Select the **Plugins** tab.
5. Fill in the plugin entry, providing a unique name, a version, and the URL to the Viz Mosart NRCS Plugin (*copied* in step **1**).
6. Setup is complete.
   When editing a selected rundown, the user right-clicks for **New > Create using** <given name of the Plugin>

For more details, see the *Viz Mosart Administrator Guide*, section **Manus Administrator > Settings Editor MOS > NRCS Configuration**.

Note: Plugin authentication is not currently supported with Octopus.

OpenMedia

Connections to OpenMedia must be setup as a project with OpenMedia.
Please contact OpenMedia for further details.

Info: Both OpenMedia hosting modes are supported. OpenMedia 5.5 and later embeds the plugin in a WebView2 control and exchanges MOS messages through it; earlier versions host it as a plain HTML plugin. The plugin detects which of the two it is running in, so there is nothing to configure for this.

Copyright © 2026 Vizrt

Showmaker

See section Setup and Administration of Showmaker in this guide.

> **Info**: No separate NRCS setup is needed for Showmaker. It embeds the plugin itself, as one of the integrations in its Assets panel, and again on its **Show Assets** page.

6.3.9 Troubleshooting

See section Troubleshooting.

Copyright © 2026 Vizrt

7 Timing Display

Note: Timing Display has been retired. The Timing Display web app was removed from Mosart Web Apps 3.1.

- All capabilities are now provided by **Smart Panel**, which is the recommended solution going forward.
- To convert existing Timing Displays, see Moving from Timing Display to Smart Panel below.
- For Mosart Web Apps 3.0 or earlier, use the Timing Display documentation for your version, as described under Documentation for earlier versions.

7.1 Moving from Timing Display to Smart Panel

Smart Panel covers everything the Timing Display did, and adds control surfaces the Timing Display never had: buttons that fire Viz Mosart commands, audio faders, live video, graphics lists, and the ability to output a whole panel as an NDI source. Rather than only timing, a single panel can now host both timing information for the studio floor and controls that operator needs.

Everything you built a Timing Display for has a direct equivalent:

In the Timing Display	In Smart Panel
Timer fields driven by the active rundown	The **Timer** item, which shows a live timer or countdown driven by Viz Mosart. The **Label** item does the same for text values.
Editing grid for laying out timers	The panel canvas. Drag an item to move it and drag a corner or edge handle to resize it; items snap to the grid set under **Grid settings** on the **Panel** tab of the Properties pane.
Drag and drop timing details from a list	The **Components** palette on the left of the editor. Every placed item also appears in the searchable **Layers** list below it.
Text formatting: color, bold, alignment, show label	Item properties: **Text color**, **Font weight**, **Text alignment** and the label settings, plus a background color with optional transparency.
Customizable audio countdown	**Audio Countdowns** in the editor. The same eight events are available: End of Current Story, Start of Next Break, End of Rundown, End of Current Package, End of Current Break, End of Current Voice Over, End of Current Float/Adlib and End of Current Item, each with a configurable offset before or after the event.

Copyright © 2026 Vizrt

In the Timing Display	In Smart Panel
Viz Mosart server status field	The **Connection Status** widget, which shows the server connection state including the main and backup beacon.
View Mode and Edit Mode	The editor and **Display mode**. Use the browser's own full-screen mode for an unattended screen.
Several named displays for different positions	Several panels. Each is listed on the Smart Panel landing page and opens independently.
Distributing a display with *Copy link*	**Copy link** on the panel card, shared the same way.
Browsing on a tablet or phone	Unchanged. A panel opens on any device with a browser.

7.1.1 Rebuilding your displays

Info: Existing Timing Display layouts are not converted automatically. A Timing Display layout is a flat list of timer fields, whilst a Smart Panel layout is a canvas of positioned items, so there is no meaningful automatic mapping between them. Saved Timing Display layouts are no longer read, the panels you want must be built once-only in Smart Panel.

In practice this takes a few minutes per display. To shorten it further, Smart Panel ships with a set of ready-made default panels that cover the usual Timing Display arrangements: pick one in the panel picker when you create a new panel, then adjust it rather than starting from an empty canvas.

Your audio countdown files do not need to be moved. Smart Panel reads them from the same media folder on the Mosart Web Apps host that the Timing Display used, so files already placed there remain available.

7.1.2 Note for Administrators

A **TimingDisplay** configuration section is still present on the Mosart Web Apps host, and its service endpoints remain in place. This is expected: Smart Panel uses them for panel layouts, images, audio countdown files and the media folder setting, along with the TriCaster credentials used for live video. Keep this configuration in place. Only the Timing Display web app itself has been removed.

Copyright © 2026 Vizrt

7.2 Documentation for Earlier Versions

If you are still running Mosart Web Apps 3.0 or earlier, the Timing Display documentation for your version remains published at the Vizrt Documentation Center. Select your version there, for example the the 3.0 User Guide, and open its Timing Display section.

Copyright © 2026 Vizrt

8 Rundown Viewer

The Mosart Rundown Viewer provides a convenient web browser view of the current Viz Mosart rundown. It is a read-only display: it shows what Viz Mosart is doing and sends nothing back, so it is safe to leave open on any screen.

- Displays the active Viz Mosart rundown, story by story, with each story's items and timings.
- Marks which items take automatically, and scrolls straight back to the story on air.
- Can show the story script, the timeline, or the Viz Mosart keyboard shortcut layout, either alongside the rundown or on their own.
- Supports a variety of devices, for example an iPad or a networked phone, and works as an unattended gallery or studio-floor display.

Info: For news-breaking details on the **Mosart Rundown Viewer**, please refer to the *Mosart Web Apps Release Notes* for your version, at the Vizrt Documentation Center.

This section describes:

- Working with the Rundown Viewer
- Setup and Administration of RV

Copyright © 2026 Vizrt

8.1 Working with the Rundown Viewer

The Mosart Rundown Viewer provides a convenient, browser-based display of a Viz Mosart rundown.

Info: To browse a rundown in the Rundown Viewer, a standard Viz Mosart setup must first be running with a loaded rundown.

- Opening the Rundown Viewer
- Key Features
  - Viz Mosart server status
  - Choosing a studio from inside the app
  - Rundown display
  - Timers
  - Help and account
  - Keyboard shortcuts panel
  - Choosing what to display
  - Interface settings
- Session and Sign-in

8.1.1 Opening the Rundown Viewer

You launch the Rundown Viewer from the Mosart Web Apps Configuration Tool.

1. In the Configuration Tool, select **Home** and then the **Rundown Viewer** card.
2. The Rundown Viewer opens in your default web browser.

On a site with several studios, **Home** shows a strip of studio chips and groups the cards under *Apps for* the selected studio and **Cross-studio**. The Rundown Viewer is studio scoped, so pick the studio chip first and the card launches against that studio. See Studios.

8.1.2 Key Features

Viz Mosart server status

The Viz Mosart server status is displayed in the upper right corner.

Status	Description
Active	The Viz Mosart server is active.
Idle	The Viz Mosart server is idle.
Down or unreachable	Manus Administrator is not running, or the Mosart Remote Control Service (RCS) is not running, or it cannot be reached.

Copyright © 2026 Vizrt

Status	Description
Main and backup	Two indicators are shown when Viz Mosart is set up with a main and a backup server, and the backup has been configured in the Configuration Tool under Studios.

The indicator tells the two kinds of trouble apart. A grey beacon means the server answers but its automation is stopped or still starting, and its tooltip reads *Only Remote Control Service running*. A red beacon means nothing answered at all, and its tooltip reads *Server unreachable*: the Remote Control Service is not answering, or the network or proxy cannot reach it. If Mosart Web Apps itself stops reporting, both beacons grey out and their tooltips read *Status unavailable*, which says the status feed is gone rather than blaming the servers. Hover a beacon to read which of these it is, together with the Viz Mosart Server version where that is known.

By default the indicator shows the hostname, IP address or FQDN configured for the studio's main and backup servers on the Studios page. In newer versions of Viz Mosart, the *Server Description* configured on each Viz Mosart server under Manus Administrator settings is used instead.

Choosing a studio from inside the app

Where studios with a Viz Mosart back end have been defined in the Configuration Tool, the top bar carries a studio picker, between the rundown name and the break timer. It is there even when only one such studio is defined, and it is absent only where none has been defined. It shows the studio the viewer is connected to, or **(no studio)** when none has been resolved, and opens a list of the available studios.

Choosing one reloads the page against that studio, so the picker and the `?studioId=` parameter reach the same place. If the studio named in the address bar is not one of the studios in the list, for example after it was renamed or removed, the viewer connects to the first studio in the list instead, and the picker shows the studio it actually connected to. Only studios whose back end is a Viz Mosart server are listed, since those are the ones with a rundown to show.

Rundown display

The rundown is shown story by story, in running order, with each story's items along its row. Items that do not fit across the row continue on a new line within the same story, and the row grows to hold them. A single item longer than the row is cut off at the right-hand edge.

- **Zoom. Zoom in** and **Zoom out** in the left toolbar widen or narrow the item columns in a story row, which is what makes the full text on an item readable. Each press steps the zoom, and the buttons stop at the ends of the range.
- **Scroll to on-air story.** Whenever the story on air has been scrolled out of view, a **Scroll to on-air story** button appears over the rundown; clicking it jumps straight back to it. It is not shown while the on-air story is already visible.
- **Autotake indicator.** An item that is set to take automatically is marked with a vertical line and an arrow at the end of its planned-duration bar, matching the autotake marker in the Viz Mosart rundown window. This tells you at a glance which items advance by themselves and which wait for the operator.
- **Story script. Show Story Script** in the left toolbar opens a script panel alongside the rundown, and **Hide Story Script** closes it again. The panel shows the script for the story on air together with the story coming next. Click a story in the rundown to highlight it and show its script in place of the next one, until the next

Copyright © 2026 Vizrt

story changes. Click it again to go back. The script can also be opened on its own, or alongside the rundown from the start, with the parameters under *Choosing what to display* below.

Timers

A timer bar below the top bar carries the rundown's timing information: **OVER/UNDER** for the whole rundown, **STORY COUNTDOWN** for the story on air, and **ITEM COUNTDOWN** for the item on air. A value that starts with a plus sign means the timer is over, and its panel turns dark red. The time to the next break is shown as **Next break** in the top bar itself, and the Viz Mosart server status sits at the right-hand end of the timer bar.

Help and account

The **Need help?** button at the bottom of the left toolbar opens a menu with **Documentation**, **What's new**, **Give feedback** and **About**, the last of which reports the Rundown Viewer version and build.

Where authentication is enabled, the signed-in user's avatar sits at the right-hand end of the top bar. Clicking it shows the account and offers a way to sign out.

Keyboard shortcuts panel

The Rundown Viewer can display the Viz Mosart keyboard layout, showing which shortcut is mapped to which physical key, including the navigation keys and the numeric keypad unless you hide them. It is a reference display rather than a control surface: use it to see and to teach the keyboard mapping in use.

Show Keyboard in the left toolbar opens the panel, and **Hide Keyboard** closes it again. The panel takes up to half the screen alongside the rundown. To fill the whole window with it, for example on a dedicated monitor in a training room, open the Rundown Viewer with the `keyboard` parameter described below.

The settings button at the top right of the panel, or a right-click on it, opens **Keyboard panel settings**. **Arrangement** switches between **Keyboard** and **Grid** (the button panel of the Viz Mosart GUI), and **Navigation keys** and **Numeric keypad** show or hide those blocks. **OK** saves your choice in this browser. To fix the arrangement for a display, add `layout=grid`, `navKeys=false` or `numpad=false` to the URL, for example `?keyboard&layout=grid`. The URL overrides the saved choice.

Choosing what to display

The Rundown Viewer can be opened showing only part of its content, which is what makes it useful as a wall or gallery display. Add one of these parameters to the URL:

Parameter	What you get
`?show-story-script`	The rundown together with the story script.
`?story-script`	The story script only.
`?timeline`	The timeline only.

Copyright © 2026 Vizrt

Parameter	What you get
`?keyboard`	The keyboard shortcut panel only, filling the window.
`?studioId=<studio>`	Connects to a particular studio's Viz Mosart server on a multi-studio site.

Tip: Because each of these is just a URL, a browser in kiosk or full-screen mode pointed at one of them makes a self-contained display that needs no operator. Several screens can show different views of the same rundown at the same time.

Interface settings

The **Interface settings** button in the left toolbar opens a dialog headed **Interface settings**. It controls how much of each story title is shown, so the rundown can be tuned to the screen it is on. The settings are remembered in the browser you set them in, so each display can differ.

Setting	What it does
Max lines for story title	A segmented control offering 1, 2, 3, 4 and **4+**. **4+** is the default and means no limit. Fewer lines fit more stories on screen; more lines make long titles readable.
Full story title when CUED	Shows the complete story title for the cued story, whatever the line limit.
Full story title when ON AIR	Shows the complete story title for the story on air.

Save applies the changes and **Cancel** discards them.

Showing complete titles throughout is not part of this dialog. It is the **Expand story title** and **Collapse story title** toggle in the left toolbar, and it is unavailable while **Max lines for story title** is set to **4+**, because there is nothing left to expand.

Tip: For a wall display, keep **Max lines for story title** low so more of the rundown fits, and turn on **Full story title when CUED** and **Full story title when ON AIR** so the two stories that matter most are always fully readable.

8.1.3 Session and Sign-in

This applies only where authentication (OIDC) is enabled for Mosart Web Apps.

The Rundown Viewer keeps its session alive for as long as its page is open, so a viewer on a gallery wall or a producer's desk is never signed out for sitting still.

It can still be signed out by an outage. If the browser cannot reach the server for longer than the configured session length, a **Session timeout!** dialog appears reading *The connection to the server was lost for too long, so you have*

Copyright © 2026 Vizrt

been logged out., with a **Login** button. Opening the Rundown Viewer against a session that had already lapsed, for example on a display PC that was switched off overnight, shows the same dialog reading *Your session has ended. Log in to continue.* instead.

An administrator can choose to let the Rundown Viewer time out when idle. Where that has been done, a **You will be logged out soon** dialog appears about a minute and a half before the session ends, with a countdown and the choice of **Continue session** or **Log out now**. See Setup and Administration of RV.

Tip: For an explanation of the elements in a Viz Mosart rundown, see the Viz Mosart User Guide, under **User Interface**, the section **Rundown Window**.

Copyright © 2026 Vizrt

8.2 Setup and Administration of RV

- Setting up the Rundown Viewer
  - Prerequisites
  - Installation and Configuration
  - Access
  - Multiple Studios
  - Setting up an unattended display
- Optional Server Setups
  - Support for main and backup connection
  - Support for HTTPS
  - Session Timeout
- Troubleshooting

8.2.1 Setting up the Rundown Viewer

Prerequisites

- A running Viz Mosart system, with *Manus Administrator* running and the *Viz Mosart client UI* running and displaying a rundown.

> **Info**: Individual Rundown Viewer features depend on the Viz Mosart server supporting them, for example the autotake indicator. For the Viz Mosart version required by your Mosart Web Apps release, see the *Mosart Web Apps Release Notes* at the Vizrt Documentation Center.

Installation and Configuration

The Rundown Viewer needs almost no configuration of its own. It reads the shared Mosart server settings that the other Mosart Web Apps use. Its one setting of its own is whether its session may time out, described under *Session Timeout* below.

1. Ensure your Viz Mosart system is operating under normal working conditions, as above.
2. From the Mosart Web Apps Configuration Tool, select **Home** and then the **Rundown Viewer** card.

Access

The Rundown Viewer has no roles of its own. Where authentication is enabled for Mosart Web Apps, users sign in as they do for the other apps, but there is no separate viewer or author permission to grant for this app: anyone who can reach the URL can read the rundown. It is a display, so it sends no commands to Viz Mosart.

Multiple Studios

On a site that runs several Viz Mosart back ends from one Mosart Web Apps host, the Rundown Viewer shows the rundown of one studio at a time.

- Define the studios once in the Configuration Tool. See Studios.

Copyright © 2026 Vizrt

- Launch the Rundown Viewer for a studio from the Home page, or open it with `?studioId=<studio>`.
- Only studios whose back end is a Viz Mosart server are offered, since those are the ones with a rundown to show.
- An operator can also switch studio from inside the Rundown Viewer, using the picker in the top bar. It appears whenever any studio is configured, not only on a multi-studio site, and choosing a studio reloads the page with that studio's `?studioId=`. An unattended display therefore stays on the studio it was pointed at across reloads.

Setting up an unattended display

The Rundown Viewer is often put on a gallery or studio-floor screen with nobody driving it. Because each view is just a URL, that needs no special deployment:

1. Decide what the screen should show, and add the matching parameter to the URL. The options are listed under Choosing what to display: the rundown, the rundown with the story script, the script alone, the timeline alone, or the keyboard shortcut panel alone.
2. Point a browser at that URL in kiosk or full-screen mode.
3. Set the display options once on that screen. They are stored in that browser, so each display keeps its own settings and you can tune one screen without affecting the others.

> **Info**: Because the settings live in the browser, replacing the machine or clearing its browser data resets that screen to the defaults. Note down the settings you chose for any screen you would not want to reconfigure from memory.

8.2.2 Optional Server Setups

Support for main and backup connection

- If you have a *backup server* for redundancy, define the environment on the Studios page, where each studio carries its own main server and optional backup. The Rundown Viewer then shows an indicator for each of the two servers.

Support for HTTPS

- If you are using *HTTPS*, define the environment in HTTPS and Certificates.

Session Timeout

Where authentication is enabled, the Configuration Tool's **Server Configuration** page carries an **Allow session timeout** group inside **Authentication (OIDC)**, with one switch per web app.

- The **Rundown Viewer** switch is *off* by default, reading **Stays signed in while open**. The viewer renews its own session for as long as the page is open, so an unattended display never drops to a login screen mid-show.
- Turning the switch on makes it read **Renews while in use; can time out when idle**, which is the right choice on a shared machine where an abandoned session should not stay signed in.

Copyright © 2026 Vizrt

- **Session timeout**, beside the switches, sets how long an unused session survives, in minutes, from 5 to 10080, defaulting to 30. It applies only to the apps whose switch is on. A value outside the range is clamped, and one that is not a whole number falls back to 30; both are logged at startup.

> ⚠ **Warning**: Leave the switch off for any gallery or studio-floor screen. Nobody touches an unattended display, so an idle timeout would put a login dialog over the rundown in the middle of a show. See Authentication.

8.2.3 Troubleshooting

See section Troubleshooting.

Copyright © 2026 Vizrt

9 Showmaker

9.1 About Showmaker

Showmaker is a lightweight, web-based, efficient MOS-rundown creation tool, eliminating the need of a complex newsroom system.

Showmaker provides all features needed to quickly create and control a Viz Mosart show, with rundowns and stories that contain automation templates, graphics, clips and scripts.

9.1.1 Typical Usage

- Standalone creation of Viz Mosart-automated shows.
- Provide an independent backup for a newsroom system.
- Quickly get on air. Especially useful at the beginning of a project:
  - The automation is prepared in advance, not requiring any newsroom system.
  - If you are using a newsroom system, all automation can be prepared upfront in Showmaker.
- Enables non-journalist skeleton staff (for example, in the middle of the night) to get breaking news on-air.

Info: Showmaker is not intended as a replacement for a newsroom system. Showmaker offers self contained, quick and easy creation and management of a Viz Mosart rundown.

9.1.2 Key Features

- **Shared production:** Team members collaborate in a single place, preparing and managing rundown content for automated or semi automated productions. You can see who else is in a rundown while you work, leave comments on a story, and draw a colleague in with an **@mention** that reaches them in their own inbox.
- **Structured workspace:** Visually organize with a clear overview, shows, rundowns and rundown templates, all in one place, and jump straight back to the rundowns you edited most recently without going through the show that holds them.
- **Automation integration:** Add and edit Viz Mosart templates.
- **Graphics gathered:** Optional Viz Flowics (HTML graphics) and Viz Pilot Edge graphics all in the same tool.
- **Show assets:** Keep a show's recurring graphics in one place and drop them into a story when needed, either with **Insert show asset** on an item or by typing a slash command straight into the script.
- **Story crafting:** Rapid modification of scripts, manual items, and timing information in a single, elegant UI. Item durations can be calculated from the script itself, using your own reading-speed presets, and a story's duration follows the items inside it.
- **Editorial metadata:** A story can carry properties as well as automation, either a value chosen from a list the show defines or a free-form name and value typed on the story, plus a synopsis. Give a value a colour and every story carrying it shows a coloured chip in the story list, and filter the list down to just the stories you need.

Copyright © 2026 Vizrt

- **Story library and approval:** Write a story before its rundown exists, plan it for a day, and place it in a rundown when the running order is ready. Every story is Draft, Ready or Approved, and an approval lapses when the story's title, script or items change.
- **Show settings in one place:** What new rundowns in a show start from, its reading-speed presets, its story properties and the automatic clearing of its old rundowns are all configured per show, on a single **Show settings** page that saves as you go.
- **Video clips:** Search the media server by clip name, preview a clip at full resolution before committing to it, and, where a TriCaster is used, upload a clip to its clip server from inside Showmaker.
- **Synchronized operations:** Updates display immediately in the active Viz Mosart rundown.
- **Confidence before air:** An attributed change history shows who changed what in a rundown and when, and an undo step recovers a story or item deleted by mistake.
- **Controlled access:** Where authentication is enabled, the people who build shows and the people who only need to read them can be separated by role.
- **Future-proofed**: Modern, secure, web-based architecture.

9.2 Why Showmaker

Traditional newsroom systems are powerful but complex, with long onboarding, heavy configuration, and far more functionality than many productions actually need. Showmaker takes the opposite approach. Light, focused, and built around the small set of tasks needed to drive a Viz Mosart rundown: open a show, build stories, add automation and graphics, go on air. New users are productive in minutes rather than days, a practical choice for teams wishing to move away from a heavyweight NRCS or that need a fast, self-contained way to run shows.

Showmaker is also designed to be the operator's own workspace. The show banner, with its image, title and program window, makes each show immediately recognizable for the people who run and own the content. A fast and focused tool that operators can familiarly return to.

9.2.1 Typical Workflow

Showmaker takes you from a blank screen to a ready-for-air rundown in five short steps:

1. Create a show.
2. Build the rundown.
3. Populate the stories with Viz Mosart templates and other instructions.
4. Activate the rundown so that it syncs with Viz Mosart.
5. Mark it ready to air, so that it becomes the operator's active rundown.

9.3 Showmaker in your Ecosystem

Showmaker is part of the suite of *Mosart Web Apps*, browser-based tools served from a single host and included with every Viz Mosart delivery.

Showmaker covers the role played by a newsroom system in a traditional setup, owning the rundown and pushing it to Viz Mosart for playout.

Copyright © 2026 Vizrt

Component	Role for Showmaker	Required?
Viz Mosart Server	Receives the Showmaker rundown (over MOS, with Showmaker acting as a virtual NRCS) and plays it on air. Also the source of the Viz Mosart automation template sets that Showmaker displays in its **Assets** panel.	Yes
Mosart Web Apps Configuration Tool	Required for one-time setup: - Point Showmaker at the Viz Mosart server or servers. - Point Showmaker at any optional graphics integrations. See Setup and Administration of Showmaker.	Yes, once only
NRCS Plugin	A Mosart Web App that Showmaker embeds, to read Viz Mosart template sets and to import data from a connected NRCS.	Yes, bundled
Viz Pilot Edge	Showmaker embeds a Pilot Edge panel, for editing Pilot Edge graphic templates in place.	Optional
Viz Flowics	Showmaker embeds a Flowics panel, for real-time cloud-based HTML5 graphic editing.	Optional
Prompter	The Mosart Web Apps prompter reads the scripts written in Showmaker, and can follow the rundown as it plays.	Optional

Technically, Viz Pilot Edge, Viz Flowics and the Mosart NRCS Plugin display asset panels inside Showmaker as iframes that exchange MOS messages directly in your browser, so you do not have to open separate applications.

Copyright © 2026 Vizrt

9.4 Getting Started

Here you learn to open Showmaker, recognize the main views, understand some important terminology, then create and run your first rundown.

- Opening Showmaker
    - Authentication
- Tour of the Interface
    - Getting back to recent work
- Key Terms
- Show Settings
- Your First Rundown
- Getting Help

9.4.1 Opening Showmaker

First try this

Showmaker runs in a web browser. Enter the URL your administrator provided. It will look something like this:

```
https://<your-mosart-web-apps-host>/showmaker
```

Otherwise, launch the Mosart Web Apps Configuration Tool

- On the Web Apps server, use the Configuration Tool's desktop or Start menu shortcut. From any other machine, open `/configtool` on the Mosart Web Apps host in a browser, if your administrator allows it.
- From **Home**, select the **Showmaker** card. On a multi-studio site Showmaker is grouped under **Cross-studio**.

Authentication

Whether you need to sign in depends on how your administrator has configured Mosart Web Apps.

- **Authentication off**: you reach Showmaker without signing in. This is the default.
- **Authentication on**: you are redirected to your organization's identity provider and sign in there. Once in, your email appears in the user avatar at the top right.

> **Info**: Showmaker is one of several Mosart Web Apps, and the same host serves them all. Switching between them does not require signing in again.

Being signed out when idle

Where authentication is on, a Showmaker session ends after a stretch with no activity. The window is thirty minutes unless your administrator has changed it, and the clock runs from your last activity at the keyboard or mouse, so nobody who is working is signed out mid-sentence.

Copyright © 2026 Vizrt

Ninety seconds before the end, a dialog headed **You will be logged out soon** counts down and offers **Continue session** and **Log out now**. If the countdown runs out you get **Session timeout!** and a **Login** button.

This applies to Showmaker, the Template Editor and the Named Overlays Editor. Smart Panel, the Prompter, the Rundown Viewer and the NRCS Plugin stay signed in for as long as their page is open, because they are on-air surfaces. Your administrator can change either decision; see Server Configuration.

Roles

Your role decides what you can do. When authentication is on, Showmaker reads your role from the identity provider:

- **showmaker-admin**: full access to create, edit, activate and put rundowns on air.
- **showmaker-viewer**: read-only access to browse shows, rundowns and templates. Editing controls are hidden or disabled.

> **Info**: If you sign in successfully but are told you do not have access, you have been authenticated but not granted either Showmaker role. Ask your administrator to add one in your identity provider. If instead you are told your access could not be checked, the check itself failed rather than being refused, and you can retry.

Troubleshooting

If Showmaker opens but you cannot see any shows, or the rundown table cannot reach a Viz Mosart server, the one-off configuration is still needed. See Setup and Administration of Showmaker.

9.4.2 Tour of the Interface

Showmaker has six views:

- **Landing page**: select a show and browse its rundowns or rundown templates.
- **Rundown page**: build and edit a single rundown.
- **Rundown template page**: build and edit a single rundown template.
- **Story library**: the show's stories that are not in a rundown yet, on the **Stories** tab of the landing page.
- **Show Assets page**: the show's saved graphics, reached from the **Assets** button on the show banner.
- **Show settings page**: how the whole show behaves. Administrators only. See *Show Settings* below.

> **Info**: All of them are described in detail in Working with Showmaker.

Getting back to recent work

On the rundown page, the button next to **Home** opens **Recently edited rundowns**. The panel is headed **Recently edited** and lists up to eight rundowns, most recently opened first, leaving out the one you already have open. Each row gives the rundown's name with its show's name, a **Template** tag where the entry is a rundown template, and when it was last edited. Click a row to open that rundown, switching show if it belongs to another one. Before you have opened anything else the panel reads *Other rundowns you open will appear here*.

The button is on the rundown and rundown template pages only, not on the landing page, the Show Assets page or the Show settings page, because those are show-level pages rather than places you switch rundowns from.

Copyright © 2026 Vizrt

Where authentication is on the list belongs to your account and follows you to another machine; with authentication off it is kept in the browser you are using. Deleted rundowns drop out of it on their own, and a renamed rundown shows its current name.

Recent timestamps here and in the rundown library read **Today at 1:03 PM** or **Yesterday at 1:03 PM**, and anything older keeps the full date, for example Jun 10, 2026, 1:03 PM. The labels roll over on their own when the local day changes, with no reload.

9.4.3 Key Terms

Showmaker organizes content hierarchically. A *show* contains *rundowns*, each rundown is made of *stories*, and each story holds the *items*, *notes* and content that go on air.

Term	Meaning
Show	Top-level container. Holds the rundowns and rundown templates for a single programme or production.
Rundown	An ordered sequence of stories that Viz Mosart can run on air.
Rundown template	A reusable rundown structure. A rundown created from a template is independent of it.
Story	A slice of a rundown. Holds the automation items, manual items, graphics, clips and scripts that share a context.
Story item	One automation step inside a story, that is a Viz Mosart template instruction.
Manual item	A story item taken to air by the operator during the show, rather than executed automatically.
Story note	A free-text block pinned to the top of a story. Not sent to Viz Mosart. Can be dragged under an item to become that item's script.
Script	Presenter or editorial text attached to a specific item. Sent to Viz Mosart with the rundown.
Graphic overlay	A Viz Pilot Edge or Viz Flowics element attached to an item.
Clip	Video media attached to an item. Only compatible story item types accept one, for example *Package* or *Voice over*.
Library story	A story written in the show's library, before it belongs to any rundown. It can carry the day it is planned for, and is placed into a rundown when the running order is ready.

Copyright © 2026 Vizrt

Term	Meaning
Show asset	A graphic saved to the show for reuse, so recurring graphics do not have to be rebuilt story by story.
Story property	Editorial metadata on a story, such as its status or its reporter. A show can define properties and the values a story may pick from, and a story can also carry its own free-form ones. Values show as chips in the story list, and the list can be filtered by them.
Reading-speed preset	A named reading speed, usually per presenter, used to calculate an item's duration from its script.
Comment and mention	A note left on a story, optionally addressed to a colleague with **@**. Unread comments and mentions are counted on the comments button in the rundown's top bar, and the panel it opens lists mentions from other rundowns separately.
Studio	A label grouping one or more MOS devices, for example the main and backup Viz Mosart servers of one gallery, or several galleries on one site. Assigned per show, and overridable per rundown.
Active	Toggle on a rundown. When on, the rundown is synced live with Viz Mosart and appears in the Viz Mosart UI.
Ready to air	Toggle on a rundown. When on, the rundown appears in the *Ready to Air* rundown listing in Viz Mosart, where the operator can take it on air.
ON AIR	A chip on a rundown when Viz Mosart is currently running it.
Floated story	A story marked to be skipped during playout. Drawn with diagonal stripes in the story list.

9.4.4 Show Settings

Everything that decides how a whole show behaves is on a page of its own, **Show settings**, rather than in the show dialog. Open it with the settings glyph in the top bar, or press **P**. The button is in the top bar of the landing page, the rundown page, an open library story and the Show Assets page. It appears only for administrators, and only once a show is selected.

The page is headed **Show settings** with the show's name beside it, and has five sections:

- **DEFAULT OPTION FOR NEW RUNDOWNS**: **Blank rundown** or **Auto-select template**. Auto-select stays unavailable until the show has at least one rundown template, and the page says so with a link to the **Templates tab**. Choosing it reveals **Set default template**.
- **Auto-calculate planned duration from script**, the setting step 5 of the walk-through below depends on, together with the show's **Reading-speed presets**. A show may hold up to four presets.
- **Auto-calculate story duration**, and which item types count toward the total.

Copyright © 2026 Vizrt

- **Story properties**: the properties this show offers its stories, and the values each one allows. A value can be given a colour, and every story carrying that value then shows a coloured chip in the story list.
- **Automatically delete old rundowns**: off for every show until an administrator turns it on, so it changes nothing for a show you have just created.

There is no Save or Cancel. Changes save as you make them, and the top bar reports the state.

Back, the left arrow at the top left or the shortcut **B**, returns you to wherever you opened the page from, and to the show library when you opened it directly by address. A separate **Home** button sits beside it only when Back would take you somewhere other than the library, so the two are never offered for the same trip. If the rundown or template you came from was deleted while you had the page open, Back stops travelling and says so, and offers a way back to the library.

Info: Nothing has to be set up here to get started. A show with no story properties defined simply has none, and Showmaker says so where they would appear. For the full detail of every setting, see Setup and Administration of Showmaker.

9.4.5 Your First Rundown

A short walk-through that introduces the core concepts. It assumes Showmaker is connected to a working Viz Mosart system, as described in Setup and Administration of Showmaker.

1. **Create a show.** On the landing page click **New show**, which is its own button in the top row of the show banner and has the shortcut **S**. Give it the title *My first show*, choose a **Studio**, and create it. Choosing a studio matters for steps 6 and 7, because the on-air controls stay disabled while a rundown has no studio.
2. **Create a rundown.** With the show open, create a new rundown, title it *Demo*, and set a planned duration of 10 minutes. Creating it takes you to the rundown page.
3. **Name the first story.** A story already exists. Click *Untitled* and rename it *Intro*.
4. **Add an automation item.** From the assets panel on the right, open the Viz Mosart templates if they are not already showing, then drag a template onto the story. Alternatively, select the template and click **ADD TO STORY**.
5. **Add a script.** Click the script area on the item and type a line or two. If your show has script timing turned on, the item's planned duration is calculated from the reading speed as you type.
6. **Sync it with Viz Mosart.** Turn on **Active** for the rundown. It is now live-synced, so it appears in the Viz Mosart UI and every edit you make is reflected there immediately.
7. **Offer it to the operator.** Turn on **Ready to air**. The rundown appears in the *Ready to Air* rundown listing in Viz Mosart, where the operator can take it on air.

Info: Both toggles are administrator-only, and there are two places to reach them: the **Active** and **Ready to air** columns of the rundown list on the landing page, and the rundown's options menu beside its title on the rundown page, where they read **Set to active** (**A**) and **Set to Ready to air** (**G**). They are linked rather than independent: turning **Ready to air** on turns **Active** on with it, and turning **Active** off turns **Ready to air** off with it. The order used above works either way.

You have now built and aired your first Showmaker rundown. To go deeper into any step, see Working with Showmaker.

Tip: Deleted a story or an item by mistake? A toast offers **Undo**, and **Ctrl+Z** does the same.

Copyright © 2026 Vizrt

9.4.6 Getting Help

The **?** icon in the top-right corner of every page opens the help menu.

Option	Opens
Keyboard Shortcuts	The list of available shortcuts. See Keyboard Shortcuts.
Documentation	This user documentation.
What's new	The Showmaker release notes.
Give feedback	A feedback mail to the Showmaker team.
About	Showmaker version and build information. Worth including in a feedback mail.

Copyright © 2026 Vizrt

9.5 Working with Showmaker

You make and organize your shows in the **Landing page** and the **Rundown page**. For a guided first-time tour, see Getting Started.

- Common Actions
- Landing Page
    - Show
    - Studio Assignment
- Show Assets
- Rundowns View
    - New Rundown
- Story Library
    - Working in the library
    - Placing a library story into a rundown
- Rundown Operations
    - 1 Rundown information bar
    - 2 Assets panel
    - 3 Story editor
    - 4 Story list and navigation panel
- Timeline View
    - Switching views
    - Reading the timeline
    - Changing timing on the timeline
    - The preview
- Story Properties
    - The Story properties panel
    - Names shared with show-defined properties
    - Property chips on the story row
    - Colours
    - Filtering the story list
- Story Approval
    - Seeing the state
    - Changing the state
- Show Settings
    - Automatically delete old rundowns
- Collaborating on a Rundown
    - Who else is here
    - Comments and mentions
    - Comments and changes
- Timing from the Script
    - Reading-speed presets
- Recording a Story
- Quick Guide to Rundown Operations
    - To add a Viz Mosart template

Copyright © 2026 Vizrt

    - To add a graphic
    - To manage a story and its items
    - To work with video clips
- Rundown Templates View
- Keyboard Shortcuts
    - Shows
    - Rundowns and templates
    - Stories
    - Items
    - Scripts
    - General
    - Not in the in-product list
- Examples, Videos, Further Guidance
    - Viz University
    - Documentation updates

9.5.1 Common Actions

Shows, rundowns, and rundown templates share the same set of management actions where they appear in Showmaker.

Action	What it does
Open	Opens the entity for editing. Double-click a rundown or rundown template row, or press Enter on the highlighted row. A show is opened with Open on the show banner, which lists the shows to pick from.
Edit	Opens a properties dialog: for a show, its title, times, banner image and Studio. A show's behaviour settings, such as what new rundowns start from and the script-timing options, are on its **Show settings** page instead.
Duplicate	Creates an independent copy of the entity. Rundowns and templates are duplicated within the same show.

Copyright © 2026 Vizrt

Action	What it does
Move or copy to show	(Rundowns and templates, administrators only) Sends the rundown or template to another show (`Shift` + `M`). In the **Move or copy rundown** or **Move or copy template** dialog, choose **Move** or **Copy**, then **Select destination show**. A rundown can be saved in the new show as a **Rundown** or as a **Template**. Saved as a rundown, it gets the **Rundown planned duration**, **Scheduled for**, **Starts at** and **Date** fields, filled in from the destination show's start and end times where it has them, and otherwise from the rundown's own schedule. A template always stays a template. An active or on-air rundown stays in its show, so it can only be copied; to move it, deactivate it once it is off air. Story properties carry over where the destination show has the same property and value. A moved rundown keeps its comments, change history and approvals, and anyone else who has it open stays in it and is told which show it went to. A copy is a new rundown, made the way **Duplicate** makes one. A rundown moved as a template becomes a new template and the original is deleted, so anyone who has it open is told it was deleted and taken back to the landing page.
Set as default	(Shows only) Opens this show automatically the next time you launch Showmaker.
Delete	Removes the entity. For shows, a confirmation dialog adds two safeguards, described under *Show actions* below.
Create rundown template from rundown	(Rundowns only) Saves the rundown as a reusable Template in the same show.
Show assets	Navigates to Show assets page where users can manage a list of assets related to a specific show.

9.5.2 Landing Page

Showmaker opens on the landing page. From here you pick a show, browse its rundowns or rundown templates, and decide which rundown is sent to Viz Mosart.

Show

A *show* is a container for related rundowns and rundown templates, typically one program or production. Each show can hold any number of rundowns and templates.

The **show banner** at the top of the landing page is both a functional space and an identity space. Functionally it is the show selector, content counter, and action menu. As an identity surface, the banner image, title, and program window personalize the workspace and signal ownership of the rundowns and templates beneath it.

Copyright © 2026 Vizrt

Show actions

- **Open**: Pick a show from the dropdown.
- **New show**: Create a new show.
  - Show title: Required.
  - Starts at / Ends at: (optional) When both are set, every new rundown and rundown template in this show inherits these values, and the planned duration is displayed automatically. Both can be overridden per rundown.
  - Banner image: (optional) A convenient identifier for the active show.
  - Studio: Selectable when a MOS device is configured. This is an administrator configuration task.
- **Edit show**: Modify any of the above on an existing show. An administrator can also click the show name on the banner. Resting the pointer on the show name shows the show's description, if it has one. Everything that governs how the show behaves, rather than what it is called, is on the **Show settings** page described under *Show Settings* below.
- **Set as default**: See *Common Actions*.
- **Delete show**: Deleting a show can affect every rundown and rundown template inside it, so a confirmation dialog provides two safeguards:
  - Choose what happens to the contents: delete the show with all its rundowns and templates, or move them to another show first. The show's library stories are deleted either way, and the dialog says how many.
  - Type the show name to confirm. The **Delete** button stays disabled until the name matches exactly.

Studio Assignment

Each show in Showmaker originates from a *Studio*, the physical production environment under the control of a Viz Mosart server.

Normally a rundown is associated with the show's original studio, but Showmaker allows rundowns to be shared between studios: you can point a single rundown at a different Studio without changing the show. This is useful for rehearsals and last-minute changes.

> **Info**: Studios are defined once when Showmaker is installed. See Setup and Administration of Showmaker.

Above the list of rundowns, the Studio badge tells you about the rundown's origins.

What you see	Status	Meaning
Grey-outlined studio badge	Default	All rundowns inherit this studio.
Orange badge	Overridden	This rundown overrides the show's default studio.
"Assign to studio" button	Unassigned	No studio is set. Click to pick one.

Copyright © 2026 Vizrt

9.5.3 Show Assets

Show assets are graphics saved on the show so they can be reused in any story of any rundown in that show. Instead of rebuilding the same lower third or bug edition after edition, you build it once, save it, and insert it wherever you need it.

To save a graphic as a show asset from an existing rundown

1. In a rundown, select a graphic you have added from the Pilot plugin.
2. Click **Save to show assets**. Showmaker confirms with *Saved to show assets*.

A graphic already in the library reports *"<name>" is already in show assets* instead of being added twice.

To insert a show asset into a story

1. Open the story you want to add the graphic to.
2. Click **Insert show asset** and pick the asset from the list. The search field narrows a long library.
3. Edit the graphic's text inline once it is on the story.

To manage the library

Click **Assets** on the show banner to open **Show assets page**. From here you can browse the library, add assets and remove ones you no longer need, without opening a rundown. An empty library shows *No assets added yet*. To add an asset from this page, build the item in a plugin on the right, then send it with the plugin's add button.

When the NRCS Plugin is opened from this page, its send button reads **Add to show assets** rather than **Add to Story**. In the rundown editor the same library is offered as a searchable list, from **Insert show asset** on an item for a graphic and from **Add show asset to story** in the story header for an automation template. The filter by type is on the **Show assets** page. The saved and modified dates on the Show assets page use the relative format described under *Rundown list details*.

Insert assets from the script

You can insert a show asset without leaving the script. Type / in the script and Showmaker offers the show's saved graphics inline; keep typing to filter the list, then choose one to attach it to the item that owns the script. It is offered to administrators only, and not on a story note, which has no item for a graphic to attach to.
A graphic inserted this way is timed from the script rather than from the start of the story: Showmaker takes it on air at the point in the read where the caret was, using the story's reading speed. The delay is worked out once, when the graphic is inserted, so editing the script afterwards does not move it. Change it by hand from the item's ***Take in delay****.*

9.5.4 Rundowns View

Switch between the **Rundowns**, **Templates** and **Stories** tabs, or press **V** to step through the three. The rundown view shows every rundown in the current show, with a search field that filters by title as you type, and a **New rundown** button.

New Rundown

Opens a dialog with two creation paths. The default follows the show's setting for new rundowns.

- **Start from scratch**: creates an empty rundown.
- **Use a template**: creates an independent copy of the selected template. The section is offered only once the show has at least one rundown template.

Copyright © 2026 Vizrt

Starts at and **Planned duration** are pre-filled from the show's settings. Change them here to override for this rundown only.

Scheduling a rundown for a date

A rundown can be given the date it is meant to air, so a show can hold one rundown per day without their titles having to carry the date.

Tick **Scheduled for** in the dialog to enable its two fields: **Starts at** for the time, and **Date**, which opens a calendar (*Pick a date*). The date is chosen from the calendar rather than typed. Ticking the box fills in today, so a rundown is never left half scheduled, and leaving the date empty is refused with **Choose a date.**

If another rundown in the show already has that date and time, the dialog warns: *<title> is already scheduled for this date and time.* You can still save, so a main and a backup rundown can share a slot.

> **Info**: A rundown template takes a start time but no date, so the same control reads **Add start time** there. A template is a shape to build from, not something that airs on a day.

Rundown list details

Column	Description
Rundowns	Rundown title. An **ON AIR** chip appears when Viz Mosart is currently running it.
Stories	Number of stories in the rundown.
Planned duration	Planned length of the rundown.
Scheduled for	The date and time the rundown is scheduled for. On the **Templates** tab the column reads **Start time** instead and carries the template's start time, since a template has no date.
Created / Edited	When the rundown was created, and when it was last edited.
Active	Toggle for MOS active. See *Putting a rundown on air*.
Ready to air	Toggle for ready-to-air. See *Putting a rundown on air*.

Recent dates are written relative to today: a timestamp on the current or previous local calendar day reads **Today at 1:03 PM** or **Yesterday at 1:03 PM**, and anything older keeps the full form, **Jun 10, 2026, 1:03 PM**. A date that was never set reads **--**. This applies to the **Created** and **Edited** columns on the **Rundowns**, **Templates** and **Stories** tabs, to the **Scheduled for** column on the **Rundowns** tab (an unscheduled rundown reads **-- --**), to the dates on the Show assets page, and to the **Recently edited** menu.

Copyright © 2026 Vizrt

Two details worth knowing: the labels correct themselves when the local day rolls over, so a tab left open overnight does not go on calling yesterday's rundowns today's; and sorting one of these columns uses the underlying timestamp rather than the text on screen, so a mixed column of relative and absolute dates still sorts correctly.

Putting a rundown on air

Two toggles control how a rundown reaches Viz Mosart. Use them in this order:

1. **Active**: turn on to live-sync the rundown with Viz Mosart. Your rundown appears in the Viz Mosart UI, and edits in Showmaker are reflected immediately.
2. **Ready to air**: turn on and the rundown appears in the *Ready to Air* rundown listing in Viz Mosart, where the operator can take it on air.

9.5.5 Story Library

A story does not have to wait for a rundown. The **Stories** tab on the landing page is the show's own library of stories: write one when the material arrives, and place it in a rundown once the running order is settled. An empty library says as much, with *Write a story here before its rundown exists, and place it when the running order is ready.*

A library story is a real story, with the same script, items, properties and synopsis as one inside a rundown. What it does not have is a position, because it is not in a running order yet.

Working in the library

- **New story**, shortcut **N**, asks for the **Story Title**, **Planned Duration** and **Scheduled for** day, then opens the new story. The editor is the story editor you already know, with **Back to the show's stories** to return.
- **Scheduled for** gives a story the day it is written for. It is what pairs a story with a rundown later, and the column sorts by it.
- The search field filters the list as you type, and reports *No stories match your search* when nothing does.
- In the **Stories** tab, an administrator's story menu offers **Open**, **Rename story**, **Duplicate**, **Place in rundown...** and **Delete**.
- The **Approval** column shows where each story stands editorially, and a story's menu also carries the approval entries. See *Story Approval* below.

Placing a library story into a rundown

There are two ways to place a story.

From the story. Place in rundown... (`Shift` + `P`), in a story's menu in the **Stories** tab or in the bar of an open library story, opens **Place story** in three steps:

- **Choose rundowns**: tick one or more of the show's rundowns. If the story is scheduled for a day, only the rundowns scheduled for that day are listed at first, with **See all <n> rundowns** for the rest. If no rundown is scheduled for that day, the dialog says so and lists them all.
- **Position the story**: it starts at the top of each chosen rundown's running order. Drag it, or press `Shift` + `↑` / `↓` , to where it belongs.
- **Confirm**: choose **Move it out of the library** or **Keep a copy in the library**. Placing into several rundowns always keeps the library story and puts a copy in each. Turn on **Float it on arrival** to have the story arrive floated.

Copyright © 2026 Vizrt

Showmaker then reports *Story '<name>' placed in '<rundown>'*, or *A copy of '<name>' placed in '<rundown>'*, with **Open** to go to it.

From the rundown. When library stories share a rundown's scheduled day, an administrator sees a banner in the rundown, *<n> stories are planned for this date*, with **View** and **Add all**. **View** opens **Stories planned for this rundown** with every story ticked: untick any you do not want, then confirm with **Add <n> stories**. Either way, the stories move out of the library and are added at the end of the running order.

Note: A moved story leaves the library, so it is in one place at a time and cannot drift out of step with a copy. **Keep a copy in the library** leaves two independent stories: edits to one do not reach the other.

Info: You cannot paste a story into the library. Use **Duplicate** to copy a story inside it, and **Place in rundown...** to send one to a rundown.

Note: Deleting a show deletes the stories in its library, even when you move its rundowns to another show. The confirmation dialog says how many.

9.5.6 Rundown Operations

Double-clicking a rundown opens the Rundown page. This is where you build the rundown. In the **Editor view** it has four areas, described below. The **Timeline view** lays the same rundown out for reviewing its timing; see *Timeline View* below.

1 Rundown information bar

- **Home**: back to the landing page.
- **Recently edited**: beside **Home**, a button whose tooltip reads **Recently edited rundowns**. It opens a panel headed **Recently edited** listing up to eight rundowns, most recently opened first, leaving out the one you are in. Each row reads "Show name / Rundown name", carries a **Template** tag where the entry is a rundown template, and shows when it was last edited. Click a row to open that rundown, switching show if it belongs to another one. Before anything is in the list it reads *Other rundowns you open will appear here*. For a signed-in user the list is kept per user on the server, so it follows you from one machine to another, always resolves current titles and drops rundowns that have been deleted; where authentication is switched off there is no identity to attach it to, so it is kept per browser instead. The menu is in the rundown and rundown template editor only, not on the landing page or the Show assets page.
- **Editor view** and **Timeline view**: two buttons after **Recently edited** that switch the Rundown page between its two views, followed by **Preview**, which opens the preview panel in the Timeline view. See *Timeline View* below.
- **Title and status badges**: the rundown name with **Active** and **Ready to air** badges, lit while each is on. They appear once a studio is set, on the rundown or inherited from the show, and while the rundown is on air an **ON AIR** chip takes their place.
- **Studio badge**: click to override or reset. See *Studio Assignment*.
- **Starts at**: start time, inherited from the rundown or its template.
- **SCHEDULED FOR**: the day this rundown is scheduled for. An administrator can click it to pick another day from a calendar, keeping the same start time. A rundown template has no **SCHEDULED FOR**.
- **PLANNED DUR.**: turns red when the total story duration exceeds it.

Copyright © 2026 Vizrt

- **PLANNED DUR.**, **STARTS AT** and **SCHEDULED FOR** read **Not set** when they have no value, rather than disappearing.
  For an administrator, **Not set** opens the rundown's **Edit** dialog with the cursor in the matching field: click it, or focus it and press `Enter` or `Space`. From **STARTS AT** or **SCHEDULED FOR**, **Scheduled for** is ticked for you, so the fields are enabled. For a viewer, **Not set** is plain text.
- **Options menu**: administrators only. Set to active (A), Set to Ready to air (G), Edit, Duplicate, Create template, Move or copy to show and Delete. A rundown template's menu drops the two switches and Create template.
- **Comments** and **Changes**: two buttons, one opening the rundown's comment feed with its count of unread comments and mentions, the other its change history. See *Collaborating on a Rundown*.
- **Show settings**: for administrators, opens the show's settings page. See *Show Settings* below.
- **Help menu**: the top-right **?** icon.
- **User avatar**: when signed in, next to the **?** icon. Click for your account and to sign out. This identity is shared across all Mosart Web Apps.

2 Assets panel

Holds the building blocks that you add to stories.

- **Viz Mosart automation templates**: always available. The set depends on the connected Viz Mosart server's template set.
- **Show assets**: reached from the story rather than from this panel. Use **Insert show asset** on an item for a saved graphic, or **Add show asset to story** in the story header for a saved automation template. See *Show Assets* above.
- **Viz Pilot Edge**: available when Viz Pilot Edge is configured.
- **Viz Flowics**: available when Viz Flowics is configured.

Open an integration from the narrow rail down the side of the panel, one button per integration. The button of the open integration is highlighted, and clicking it again closes the integration. The rail is one `Tab` stop: `↑` / `↓` move between the integrations, and `Enter` or `Space` opens or closes one. Each button carries the integration's name, so a screen reader announces which one it is and whether it is open.

Switching away from an integration and back does not reload it, so a graphic you are part-way through building is still there when you return.

If Showmaker cannot reach an integration, the panel shows *<name> is not available*, with *Showmaker could not reach this integration. Check that it is running, then close and reopen it.* Its button stays on the rail.

Drag the seam between the panel and the story editor to give an integration more room.

3 Story editor

The centre panel lists all stories and their content in order. Selecting a story in the story list scrolls the editor to it.

- **Saving and live sync.** All edits save automatically as you make them; there is no Save button. While the rundown is **Active**, changes are pushed to Viz Mosart over a live connection.
- **Undo.** If you delete a story or an item by mistake, a toast offers **Undo**, and **Ctrl+Z** does the same.
- **Copy and paste a story. Copy story** puts a story on the clipboard; **Paste story** appends the copy at the end of the rundown, and **Paste story as next** puts it after the current one. The clipboard, for stories and items

Copyright © 2026 Vizrt

alike, is kept by the browser: it works across rundowns and templates and in every Showmaker tab open in that browser at the same address, and it survives a reload. Signing out clears Showmaker's copy from the browser. On https:// addresses the copy was also put on your computer's clipboard, where it stays until you copy something else, and Showmaker can pick it up from there again.

Story header

- **Story title**: editable inline.
- **Auto / Manual toggle**: a per-story view filter. *Auto* hides manual items so you see only what runs automatically; *Manual* shows only manual items. Each story has its own setting, and it does not affect what is sent to Viz Mosart.
- **Planned duration**: the story's duration. This can be calculated for you; see *Timing from the Script*.
- **Options menu**: Float, the approval entries (see *Story Approval* below), Copy, Duplicate, Paste item, Paste story as next, Delete.
- **Comments**: opens the comments for this story.

Story body

The editing area inside a story, where you add and arrange the elements that control Viz Mosart templates, graphics, timing and scripts. A story can hold any combination of:

- **Automated item**: a Viz Mosart template that runs automatically.
- **Manual item**: an item the operator can select during the show. Manual items are styled differently from scheduled ones so the two are not confused, and they are listed separately.
- **Story note**: pinned to the top of the story, a free-text field for editorial context or instructions. It stays visible while you build the automation underneath, and can be dragged under any item to become that item's on-air script.
- **Script field**: presenter or editorial text attached to a specific item.
- **Graphic**: a Viz Pilot Edge or Viz Flowics element attached to an item.
- **Clip**: a video playback element from the media server, attached to an item.

Secondary items

An item can carry *secondary* items, for example an overlay graphic attached to a camera template. Secondary items have their own actions, available from the handle or by double-clicking.

Each secondary item has a **target button** that gathers its destination and timing in one place. From it you set where the graphic goes and how it comes off air:

Take-out method	Meaning
Duration	Taken out after a set time.
Item end	Taken out with its item.
Story end	Taken out with the story.

Copyright © 2026 Vizrt

Take-out method	Meaning
Open end	Taken out manually by the operator.

Overlay graphics can also have their destination changed here, without leaving the rundown editor. While you drag a secondary item, a **Drop here** slot shows where it will land, as it does for a primary item (see *To manage a story and its items*). Secondary items can be moved between the scheduled and manual tabs.

4 Story list and navigation panel

An ordered list of every story in the rundown, with controls to add, reorder, float, copy and delete.

- **Add story (+)**: appends a new empty story. Hold **Shift** to add it as the *next* story, below the selected one.
- **More options**: **Import stories** brings in stories from a prepared CSV or Excel file, mapping each row to a new story at the end of the rundown. **Paste** pastes a story from the clipboard.
- **Story options** (click the handle without dragging): **Float story** marks the story to be skipped during playout. Floated stories are drawn with diagonal stripes and are not sent to air, but stay in the rundown for reference. Also **Copy**, **Duplicate** and **Delete**, and the approval entries; see *Story Approval* below. On one of several selected stories the menu acts on all of them; see *Selecting several stories* below.
- **Drag to reorder**: drag a row by its handle. Numbering updates automatically. Dragging one of several selected stories moves them all.
- **Items counter**: the number of scheduled and manual items in the story.
- **Property chips**: the values a story has set, shown on its row. See *Story Properties* below.
- **Approval strip**: the story's editorial state, as a coloured strip beside its number. See *Story Approval* below.

Tip: Adding a story is a frequent action, so there are two ways to do it: the **+** icon in the story list header, and the button at the foot of the editor area. Both append a new empty story; the lower one also scrolls it into view.

Resizing the story list

The story list can be widened. Drag the seam down its right-hand edge, or focus the handle and step it with the arrow keys. A single click on the handle puts the list back to its original width, and its tooltip reads **Resize the story list, or click to reset its width**.

The list never becomes narrower than the width it has always had, and it cannot be dragged wide enough to squeeze out the story editor; the ceiling is lower again while a plugin panel is open. The width you choose is remembered for you in that browser, so it survives a reload. Only the story list resizes: the **Story properties** panel keeps its own width and is pushed along.

Selecting several stories

To move or act on several stories at once, select them first, in the story list or on the story cards in the editor:

- **Shift+click** selects every story from the one you last clicked to the one you Shift+click.
- **Ctrl+click** adds a story to the selection, or takes it out. On macOS, **Cmd+click** does the same.
- `Ctrl` + `Shift` + `↑` / `↓` grows or shrinks the selection one story at a time, `Ctrl` + `A` selects every story in the list, and `Escape` clears the selection.

Copyright © 2026 Vizrt

Selected stories are highlighted in the list and in the editor, and the story the editor is on keeps its own marking. Clicking inside a selected story's card makes it the current story without losing the selection, and so does stepping to another selected story with `Shift` + `↑` / `↓`. A plain click on a story outside the selection, or a step to one, selects that story alone.

With a filter on, only the stories the list shows are selected. A selected story that the filter hides leaves the selection, and clearing the filter does not bring it back.

Moving several stories. Drag any selected story by its handle and the others travel with it. The list opens one drop slot per story where they will land, and they arrive together in their running order, closing any gaps between them. `Shift` + `Alt` + `↑` / `↓` moves the selection one place up or down, closing its gaps first. As for a single story, reordering is off while a filter is on.

Acting on several stories. For an administrator, a panel at the foot of the story list, above the duration bar, shows how many stories are selected, for example *3 selected*, with **Clear** to let the selection go. Under it are the actions, which apply to every selected story, and **Select all**. When the browser window is 800 pixels tall or less, the panel leaves out **Select all** to save space; `Ctrl` + `A` or **Select all** in the story list's menu still selects every story. Each action's tooltip names the count, for example *Float 3 stories*. The menu on a selected story's handle, and in its card's header, offers the same actions in place of its usual entries.

Action	What it does to the selected stories
Float	Floats them all, including any that were not floating yet. When every selected story is already floating, the action reads **Unfloat** and unfloats them all.
Copy	Puts them on the clipboard as one block, in running order (`Ctrl` + `Shift` + `C`). Pasting the block, from a paste entry or with `Ctrl` + `Shift` + `V`, puts the stories in that same order, and the pasted stories come back selected.
Duplicate	Adds a copy of each after the last selected story, in the same order (`Ctrl` + `Shift` + `D`). The copies come back selected.
Delete	Asks once, *Are you sure you want to delete 3 stories?*, then deletes them all. `Delete`, `Backspace` and `Shift` + `Delete` ask the same while several stories are selected. A single **Undo**, or `Ctrl` + `Z`, brings them all back.

9.5.7 Timeline View

The **Timeline view** lays a whole rundown out at once, one row per story, with each story's scheduled items drawn to scale on a shared time ruler. Use it to review a rundown before air and to polish its timing: a story running longer than planned, a graphic on air over another, a script that runs past its shot.

Copyright © 2026 Vizrt

Switching views

In the rundown information bar, the two buttons after **Recently edited** switch the Rundown page between **Editor view** and **Timeline view**. Your choice is remembered in that browser, so the next rundown you open uses the same view. Both views are offered for rundowns and rundown templates; a library story always opens in the editor.

A click on a story's row selects the story, as in the story list. To work on a story in the editor, double-click an empty part of its row in the timeline or in its manual items: Showmaker switches to the **Editor view** on that story.

Reading the timeline

The table has three columns, and every row is the same height.

Column	What it shows
Stories	Each story's row as the story list shows it. The header holds the number of stories, **Zoom out** and **Zoom in**, and the **Timeline key**.
Scheduled	Under the time ruler, the story's scheduled items end to end, each one a block as wide as its duration, in its item card's colours. An item with no duration of its own is drawn short and hatched. Its graphics are thin bars along the foot of the block, in lanes by destination, starting at their take-in; sound and accessory cues are marks at their take-in. Under the blocks are the first and last words of the script read over each item, with its reading time, in yellow where the reading runs past its shot. Without a reading-speed preset to time it by, the word count is shown instead.
Manual items	The story's manual items, a compact row each, in a list that scrolls within the row. **Collapse manual items** folds the column to a count per story, and **Show manual items** opens it again. In a narrow window the column folds by itself.

Behind each story's items, a band runs to its planned duration. The band's end turns red when the items run longer than planned, and a label says how far they run over or under.

Every story is shown: the story list's filter does not apply in the Timeline view. The zoom and the folded manual column are remembered for you in that browser.

> **Info**: **Timeline key** lists every mark, colour and pattern the timeline uses, and the keys that work on it, so you can look one up without hovering over it.

Changing timing on the timeline

An administrator can adjust the rundown straight on the timeline. Each change is saved when you let go, or when you stop pressing keys.

Copyright © 2026 Vizrt

To	Do this
Change an item's duration	Drag the right edge of its block. The edge snaps to the story's planned end.
Move an item	Drag the chip at the start of its block to another place in the story, or into another story's row. The script read over it goes with it. A click on the chip opens the item's menu.
Retime a graphic	Drag its bar to change its take-in, or the bar's right end to change its length. Either way it stays inside its block. Drop the bar on another block to move the graphic to that item.
Set the story's planned duration	Drag the end of the planned-duration band. It snaps to the ends of the blocks. A planned duration set here stops the story calculating it for you, as typing one in the editor does.
Reorder stories	Drag a story by the handle on its row, as in the story list.
Edit an item in its plugin	Double-click its block, a graphic's bar, or a manual item's row. A Quick CG graphic has no plugin: double-clicking its bar switches to the Editor view with that graphic selected, where you edit it inline.

From the keyboard, on an item's chip, `←` / `→` shortens or lengthens the item by a second and `Alt` + `←` / `→` moves it one place. On a graphic's bar, `←` / `→` moves its take-in, `Alt` + `←` / `→` changes its length and `↑` / `↓` goes to the previous or next graphic in the block. On the end of the planned-duration band, the arrow keys change it by a second, and `Page Up` / `Page Down` by 10 seconds. Hold `Shift` for steps of 10 seconds, and press `Escape` to cancel a drag.

The shortcuts that add, paste, duplicate, import or delete stories, and those that act on the selected item, are off in the Timeline view, and stories cannot be selected several at a time there. Switch to the Editor view for those.

The preview

Preview, beside the view buttons, opens a panel under the table. In the Editor view the button is off, and its tooltip reads *The preview is available in the timeline view*.

- The panel shows an item's picture, with its place in the story above it and its name below. **Previous** and **Next**, under the picture, step through the rundown's scheduled items (in a small panel they show as arrows only), and clicking a block shows that item. The item on show is ringed in the timeline.
- An item with graphics and cues steps through those too. Where there is room, they are listed beside the picture on a small timeline of their own, where an administrator can retime them as on the table.
- Beside it, a second preview shows the same story's manual items. Drag the divider between the two to share out the room, or all the way across to hide the manual preview.

Copyright © 2026 Vizrt

- While the panel is open, `Space` and `Shift` + `Space` walk forwards and back through the rundown in running order, from anywhere in the view. Turn on **Include manual items in the walk-through** in the panel's top bar to take in each story's manual items as well.
- Click the picture to open it full screen, with the clip's proxy where it has one.
- Drag the panel's top edge to resize it, or click the edge to reset its height.

Whether the panel is open, its height and its manual preview settings are remembered for you in that browser.

9.5.8 Story Properties

A story can carry editorial metadata alongside its automation, so that a rundown can be read for its state and not only its running order. There are two kinds.

Show-defined properties are authored once per show and offered to every story in it: an administrator sets up a property name and the list of values a story may choose from, and a story then picks one value per property from a dropdown. See Setup and Administration of Showmaker for how they are defined.

Custom story properties are free-form name and value pairs typed on the story itself, with no show-level setup. Anyone who can edit the rundown can add them; they are not restricted to administrators.

The Story properties panel

Both kinds live in the **Story properties** panel, opened with the toggle at the right of the story list header, tooltip **Story properties**. The toggle stays disabled until a story is selected, with the tooltip *Select a story to see its properties.*

The panel header shows the story's position as **Story <n>**, its title, and a **Float** chip when the story is floated. It has two tabs:

- **Properties**: the story's **Approval** first (see *Story Approval* below), then the show-defined properties, each with its dropdown, then the custom rows. The custom section's heading carries a count of the rows that have both a name and a value.
- **Synopsis**: a plain free-text field for a brief description of the story. Unlike a property value it gets no chip on the story list.

Empty states tell you which kind is missing: with nothing defined on the show the panel reads *No properties defined at the show level*; with no custom rows yet, *No custom properties yet*, with **Add first property** to start one. Later rows are added with **Add property** and removed with **Remove property**.

Properties, custom rows and the synopsis save as you type. They are shared live with everyone else in the rundown, they travel with a story or a rundown that is duplicated, and they come back with a story restored by undo.

Names shared with show-defined properties

A custom row whose name matches a property the show already defines is flagged **Already defined by the show**. The panel's other name checks appear the same way: **Name required** for a row that has a value but no name, and **Duplicate name** for two custom rows with the same name. In each case the row is kept rather than refused, so the message is telling you to rename it rather than blocking the edit. Matching ignores case and surrounding spaces.

Copyright © 2026 Vizrt

Property chips on the story row

Each story row shows the values that story has set, as chips. A chip's tooltip reads "Property: Value", so a chip carrying only its value still says what it is. Clicking any chip opens the **Story properties** panel for that story, and a story with no values shows a dotted add control in its footer instead, tooltip **Add properties**, which opens the same panel.

At most three chips fit on a row and the rest collapse into a **+N** chip whose tooltip lists what it stands for. Which three are shown is deliberate: *coloured* values take the places under the cap and the uncoloured ones are the ones that fold into **+N**, because a colour is a deliberate mark of something to notice. The chips that are shown still sit in the order the show defines them, so promoting a value does not move it between rows.

A coloured chip always keeps its text label, so nothing depends on being able to tell the colours apart.

Colours

A show-defined value can carry a colour, so a status reads at a glance down the story list. The colour is chosen in one place only, the **Story properties** section of the **Show settings** page, from a fixed palette of ten named colours.

A colour then shows up in exactly three places: the value's own chip on the Show settings page, a dot in front of each option in the story's value dropdown, and the value's chip on the story list row, filled in that colour. Only show-defined values can be coloured: custom properties always draw plain, and the chips in the filter panel are never coloured.

Filtering the story list

The funnel button in the story list header filters the list by approval state and property value. Its tooltip is **Filter**, and a badge on it counts how many values are currently picked. It is always available, because every story has an approval state to filter by.

The panel is headed **Filter by**, with **Clear all** beside it while a filter is active. It lists an **Approval** row first, with a chip for each state: **Draft**, **Ready**, **Approved** and **Changed since approval**. Below it comes one row per property: the show-defined properties first, then one row for each custom property name in use anywhere in this rundown, whose chips are the values those stories actually typed. A custom row is distinguished only by its tooltip, which reads "<name> (custom)", so a show property and a custom property sharing a name give two rows that look alike.

Pick values as chips. Two or more values under one property match any of them, and the panel says so: *Values within a property match any of the ones you pick.* Across properties, **Match** decides: **All** requires the story to match every filtered property, **Any** at least one. The **Approval** row counts as a property here. Names and values both ignore case, so Colour and colour are one row, Red and red one chip.

A status line reads **Showing {shown} of {total} stories**, and *Filter cleared, showing every story* once the filter goes. When nothing matches you get **No stories match this filter** and the hint *Pick different values, or clear the filter to see every story. Adding a story also clears it.*

Three behaviours are worth knowing before you rely on it:

- The filter hides non-matching stories in the story list and in the editor area alike, and the selection follows it. If the selected story stops matching, the selection moves to the first story still shown.
- Adding a story while a filter is active clears the filter, so the new story is never hidden.
- Dragging to reorder is off while a filter is active. The story handle still opens the story menu, and the numbering keeps each story's real position in the rundown.

Copyright © 2026 Vizrt

Info: If an administrator removes a property, or removes one of its values, the values stories had already chosen stay in the data but stop appearing anywhere, on the story row and in the panel alike, and the list can no longer be filtered by them. Putting the value back, spelled the same way, brings them back.

9.5.9 Story Approval

A story carries an editorial state, so a producer can tell at a glance which stories have been read and signed off and which still need work. There are three states: **Draft**, **Ready** and **Approved**. Every story starts as a draft.

An approval records who gave it and when, and it stands only for the story as it was read. Once the title, the text of a script or the list of items changes, the approval lapses: the story reads **Changed since approval**, still naming who approved it and when, until someone reads it and approves it again or the change is reversed. Adding, removing or reordering items counts, and so does changing an item's clip, template or variant. Timing and metadata do not: planned and item durations, in-times, destinations, transitions, story properties and the synopsis can all be edited without lapsing an approval. Nor does editing the text in a graphic, such as the name on a lower third.

Seeing the state

- **Story list**: a narrow strip between each story's number and its body. Green with a check pattern is approved, a dashed blue band is ready, and yellow is changed since approval. A draft shows only a faint track. Hover a coloured strip for the state and, for an approval, who gave it and when.
- **Story properties** panel: the **Approval** section at the top of the **Properties** tab has a chip for each state. Under the chips, a draft reads *Not yet marked as ready*, a ready story *Waiting for someone to read and approve it*, and an approved story names who approved it and when. A lapsed approval adds the note *The title, script or items changed after this approval. Read the story again before approving it.* with an **Approve again** button.
- **Story library**: the **Stories** tab has an **Approval** column that names every state, a draft included. Sorting by it puts the outstanding work first: drafts, then ready stories, then changed since approval, with approved stories last. An open library story also shows its state as a badge in its bar, ahead of its timings.
- **Filter**: the story list filter has an **Approval** row, so you can show only the stories in a given state. See *Filtering the story list* above.
- **Changes** panel: approval moves appear in the timeline as *approved*, *marked ready* and *sent back to draft*. A lapse has no entry of its own, because the edit that caused it is already listed. **Approve again** adds no entry either.

Changing the state

Changing a story's approval needs the `showmaker-admin` role. A viewer sees the state everywhere but cannot change it. There are four ways to change it:

- Pick a chip in the **Approval** section of the **Story properties** panel.
- Use any story menu: the handle in the story list, the options menu in the story header, the menu in an open library story's bar, or a story's menu in the **Stories** tab. The menu offers only the moves that change something: a draft can be marked ready or approved directly, a ready story approved or sent back, an approved story sent back, and a changed one approved again or sent back. The entries are **Mark as ready**, **Approve story**, which reads **Approve again** on a changed story, and **Back to draft**.

Copyright © 2026 Vizrt

- Press `Shift` + `R` to mark the selected story ready, `Shift` + `A` to approve it (or approve it again after a change), or `Shift` + `D` to send it back to draft. The keys follow the same rules as the menu, and in the **Stories** tab they act on the focused story.
- In an open library story, click the badge in its bar to move the story on one step: a draft to ready, ready to approved, approved back to draft, and a changed story to approved again. Its tooltip names the step it will take.

Approving records you as the approver, at the current time. With authentication off there are no user identities, so an approval is credited to the anonymous name shown for that browser, such as Anonymous Kiwi. Sending a story back to **Ready** or **Draft** removes the approval, and its approver and time with it. Approval changes reach everyone in the rundown without a reload.

An approval belongs to one story as it was read, so it does not carry over to a copy: a duplicated or pasted story, and every story in a duplicated rundown, starts as a draft. So does every story in a rundown copied to another show, or moved there as a template; a rundown moved to another show as a rundown keeps its approvals, because it is the same rundown. A library story moved into a rundown keeps its approval, because it is the same story. A copy (**Keep a copy in the library**, or placing into several rundowns) starts as a draft.

9.5.10 Show Settings

A show's configuration lives on its own **Show settings** page. Open it from the settings glyph in the top bar, which is on the landing page, in the rundown editor, in an open library story and on the Show assets page, tooltip **Show settings (P)**, or press **P**. It is for administrators only, and only while a show is selected.

The page holds, in order: the default for new rundowns in this show, either a blank rundown or one seeded from a chosen rundown template; **Auto-calculate planned duration from script** with its **Reading-speed presets**; **Auto-calculate story duration**; **Story properties**; and **Automatically delete old rundowns**.

There is no Save or Cancel. The page saves as you go, and a line in its bar reports the state as **Saving…**, **All changes saved**, **Couldn't save changes**, or **Fix the highlighted errors to save**. The **Back** arrow, shortcut **B**, returns to wherever you opened the page from, and the home icon returns to the library.

Where the show has no rundown templates yet, the auto-select option is unavailable, with the tooltip **Create a rundown template first** and a link to the **Templates tab**.

For the full detail of every setting, including how story properties and their colours are defined, see Setup and Administration of Showmaker.

Automatically delete old rundowns

Automatically delete old rundowns clears a show's old rundowns on a schedule, so a long-running programme does not accumulate them forever. Set **Delete rundowns older than** a number of **days**, and a **Frequency** of **Daily** or **Weekly**; a weekly schedule also takes a **Day of week** and a **Time of day**. The schedule is shown and set on your own clock, whatever region you are in.

Preview opens **Rundowns that would be deleted**, listing the oldest candidates as they stand now, or *No rundowns currently qualify.*

Four things the purge never touches, which is what makes it safe to switch on:

- rundown templates;
- a rundown that is on air;
- a rundown whose planned start is still in the future, so a show assembled well in advance is not deleted before it airs;

Copyright © 2026 Vizrt

- anything newer than the window. A rundown's age is measured from when it was last edited, falling back to when it was created.

Warning: **Deleted rundowns cannot be recovered.** Use **Preview** before you leave the page.

9.5.11 Collaborating on a Rundown

Several people can work on the same rundown at once, and Showmaker makes their presence and their changes visible rather than silent.

Who else is here

The people currently in the rundown are shown as avatars. Each person gets their own colour, and viewers are distinguished from editors, so you can tell at a glance who is reading and who is working.

Comments and mentions

- Every story can carry comments. Open **Story comments** from the story, type in **Add a comment…** and send it. Replies keep the conversation together.
- Type **@** in a comment to mention a colleague. They are notified, and the comment is marked so they can see they were mentioned.
- A story with activity you have not seen is flagged, so you do not have to open each story to check.

Comments and changes

Two buttons in the rundown's top bar open two panels of their own:

- **Comments**: every comment and mention in this rundown, with unread ones counted. Mentions from other rundowns are gathered separately, so a mention cannot be missed just because you were somewhere else.
- **Changes**: an attributed timeline of what changed in this rundown, showing who did it and when. Use it before air to review what has moved since you last looked. Approvals are listed too; see *Story Approval* above.

Info: The mention picker is populated from the people who have signed in to Mosart Web Apps. With authentication switched off there are no identities to choose from, so the picker is empty.

9.5.12 Timing from the Script

A story's timing can be derived from the material itself rather than typed in by hand. Both settings are per show, on the **Show settings** page.

- **Auto-calculate planned duration from script**: when a script is added to a story item, the item's planned duration is set from the talent's reading speed.
- **Auto-calculate story duration**: each story's planned duration is set from the sum of its item durations.

Copyright © 2026 Vizrt

Reading-speed presets

Reading speed varies by presenter, so it is kept as a named list of **reading-speed presets**: a preset name, for example the talent's name, and a speed. A show may hold up to four presets, and **Add preset** disappears once the fourth exists. Pick the right preset for a script.

A reading speed must be a whole number between 60 and 400 words per minute; the field says **Use 60-400 wpm** otherwise.

> **Info**: Remove is disabled for two reasons, each with its own tooltip. On the show's default preset it reads **The default preset can't be deleted. Set another preset as default first**, so there is always a speed to fall back on. On the only remaining preset, while the feature is on, it reads **Keep at least one preset while the feature is on**.

Deleting a preset asks first. The confirmation names the preset and spells out the consequence: story items using it fall back to the show's default preset, and any durations calculated from script are recalculated. Where no default preset is set, it says instead that those planned durations stay as they are until one is.

9.5.13 Recording a Story

Where *Story Recorder* is enabled by your administrator, each story carries recording controls:

- **Record**: marks the story to be recorded by the Story Recorder feature.
- **Publish**: marks the story for online publishing. This is available with a TriCaster only.
- **Use recording**: replaces the story's script with the recorded version.

If you do not see these controls, they have not been enabled. See App Configuration.

9.5.14 Quick Guide to Rundown Operations

To add a Viz Mosart template

1. Open the Viz Mosart assets panel on the right. The coloured templates represent automation commands controlling devices such as cameras, video servers, graphics engines and audio mixers.
2. Pick a template and adjust its properties as needed.
3. Click **ADD TO STORY**, or drag the template onto a story.

To add a graphic

1. Open the panel for Viz Mosart, Viz Pilot Edge or Viz Flowics, or select **Show assets** for one you have already saved.
2. Pick a graphic and adjust its properties.
3. Click **ADD TO STORY**. The graphic is attached to the selected item.

> **Info**: Which graphics panels you see depends on how your administrator has configured Showmaker.

Copyright © 2026 Vizrt

To manage a story and its items

- Click an item's handle to reveal its actions. You can copy, duplicate, edit, add a script to, change the timing of, or delete it.
- Click the script icon to edit the script.
- Drag an item's handle to change its execution order. While you drag, the item travels as a single chip, and an item-sized **Drop here** slot opens where it will land: anywhere in a story counts, the story header puts it first and the space below the rows puts it last. Outside a story the cursor shows that you cannot drop there, and letting go cancels the move. An empty story offers its whole box as the slot. On the drop the item is selected, as if you had clicked it.

To work with video clips

- Search the media server by clip name. Results show the full clip name, and a thumbnail where the media server provides one.
- Open a clip to preview it at full resolution, with a transport bar, before committing to it.
- Where a TriCaster is used, a clip can be uploaded to its clip server from inside Showmaker.

9.5.15 Rundown Templates View

A *rundown template* defines the structure of a rundown without specific content. Use templates for recurring programmes and standard show formats. Template lists work like the *Rundowns View*, so refer to that section for how to manage them.

> **Info**: The differences, reflecting what a template is for:
> - **Template editing is local.** Once a rundown has been created from a template it is independent, so editing the template later affects only new rundowns created from it.
> - **No Active or Ready to air.** Templates are never synced to Viz Mosart and never go on air, so those badges and controls are absent.
> - **No Studio badge.** Studio assignment is a rundown property. A rundown created from a template inherits the show's studio, or whatever override is set on the rundown.
> - **Options adapted for templates**: Open, Edit, Duplicate, Move or copy to show and Delete, with no Create template entry. To build a rundown from a template, use **New rundown** and pick the template there.

9.5.16 Keyboard Shortcuts

Open this list inside Showmaker at any time with **K**, or pick **Keyboard Shortcuts** from the Help menu.

The tables below follow the same grouping as that in-product list, because the same action often works both on the landing page and in the rundown editor. Shortcuts marked **Administrator** need the `showmaker-admin` role, and are hidden from the in-product list for a user who does not have it.

Copyright © 2026 Vizrt

Shows

Action	Shortcut	Administrator
Open show	`O`	
New show	`S`	Yes
Edit show	`E`	Yes
Show settings	`P`	Yes

Rundowns and templates

Action	Shortcut	Administrator
Browse list of rundowns	`↑` / `↓`	
Focus search (Find)	`F`	
Switch rundown / templates tab	`V`	
Create new rundown	`R`	Yes
Create new template	`T`	Yes
Make active	`A`	Yes
Go - ready to air	`G`	Yes
Edit rundown	`Shift` + `E`	Yes
Duplicate rundown	`Ctrl` + `D`	Yes
Create template from rundown	`Shift` + `T`	Yes
Move or copy to show	`Shift` + `M`	Yes

Copyright © 2026 Vizrt

Action	Shortcut	Administrator
Delete rundown	`Delete`	Yes

Stories

Action	Shortcut	Administrator
Browse stories	`Shift` + `↑` / `↓`	
Select a range of stories	`Shift` + `Click`	
Add or remove a story from the selection	`Ctrl` + `Click`	
Extend the selection up / down	`Ctrl` + `Shift` + `↑` / `↓`	
Select all stories	`Ctrl` + `A`	
Clear the story selection	`Escape`	
Add story	`N`	Yes
Edit story title	`E`	Yes
Reorder story	`Shift` + `Alt` + `↑` / `↓`	Yes
Copy / Paste story	`Ctrl` + `Shift` + `C` / `V`	Yes
Duplicate story	`Ctrl` + `Shift` + `D`	
Place library story in a rundown	`Shift` + `P`	Yes
Mark story as ready	`Shift` + `R`	Yes
Approve story, or approve it again after a change	`Shift` + `A`	Yes

Copyright © 2026 Vizrt

Action	Shortcut	Administrator
Send story back to draft	`Shift` + `D`	Yes
Delete story	`Shift` + `Delete` or `Shift` + `Backspace`	Yes
Delete the selected stories (several selected)	`Delete` or `Backspace`	Yes
Import from CSV	`Ctrl` + `I`	Yes

Items

Action	Shortcut	Administrator
Navigate Items	`↑` / `↓`	
Reorder item	`Alt` + `↑` / `↓`	Yes
Copy / Paste item	`Ctrl` + `C` / `V`	Yes
Delete item	`Delete` or `Backspace`	Yes
Toggle Manual Assets	`M`	

Scripts

Action	Shortcut	Administrator
Add script to item	`S`	Yes
Edit script section	`Shift` + `S`	Yes

Copyright © 2026 Vizrt

General

Action	Shortcut	Administrator
Home / return to library	`H`	
Show keyboard shortcuts	`K`	

Not in the in-product list

These work but are not shown by **K**:

- **Undo delete**, `Ctrl` + `Z`, for administrators.
- **Open the highlighted rundown**, `Enter` or `Space`.
- **Back**, `B`, on the **Show settings** page only.
- `Insert` as an alternative to `N` for adding a story, and `Home` as an alternative to `H` (in the Editor view only).
- On the **Stories** tab, `Enter` or `Space` opens the focused story, and for administrators `Shift` + `E` renames it, `Ctrl` + `D` duplicates it and `Delete` deletes it.
- In **Place story**, `Shift` + `↑` / `↓` moves the story in the running order.

9.5.17 Examples, Videos, Further Guidance

Showmaker is evolving rapidly. Below is a starting point for finding useful descriptions online.

- A short introductory video takes you through the basics.
- Making rundowns easy and intuitive to create, with drag-and-drop access to Viz Mosart templates.
- A three-minute clip linking Viz Mosart operations to the show's rundown with newsroom tags.

Viz University

Showmaker training is offered free of charge to registered customers at Viz University.

Documentation updates

As development evolves, the documentation is continually updated. Between software updates, the latest documentation is available at the Vizrt Documentation Center.

Copyright © 2026 Vizrt

9.6 Setup and Administration of Showmaker

Showmaker needs a few one-off configurations before it is fully usable. This section covers them, both for the system administrator and for the Showmaker user.

Info: Documentation is continually updated, see the latest version at the Vizrt Documentation Center.

- System Administrator Only
- Setting up Showmaker for Daily Use
    - Prerequisites
    - Installation and Configuration
    - Studios
    - Show Settings
    - Roles and Access
    - Video Clips
    - To work with the Story Recorder feature
    - Graphics Integrations
- Optional Server Setups
    - Support for main and backup connection
    - Support for HTTPS
- Troubleshooting

9.6.1 System Administrator Only

- Follow the standard setup described in section Configuration Tool, together with the Showmaker-specific instructions, then return to this page.

9.6.2 Setting up Showmaker for Daily Use

Prerequisites

- Viz Mosart 5.14.3 or later.
- *Manus Administrator* running.
- The *Viz Mosart client UI* running and able to display a rundown.

Info: Individual Showmaker features depend on the Viz Mosart server supporting them, so for the version required by your Mosart Web Apps release, see the *Mosart Web Apps Release Notes* at the Vizrt Documentation Center.

Installation and Configuration

You can open a standalone, pre-production rundown in Showmaker with no additional configuration. To work with Viz Mosart templates, or to test the rundown against Viz Mosart, complete the one-off steps below.

Copyright © 2026 Vizrt

To manage Viz Mosart templates

To work with Viz Mosart template sets in Showmaker, the bundled *NRCS Plugin* must be correctly configured, because Showmaker embeds it to read those template sets.

- Follow the *Showmaker* instructions under App Configuration.

To connect to Viz Mosart

1. Ensure your Viz Mosart system is operating under normal working conditions. As a minimum, Manus Administrator must be running, and the Viz Mosart UI must be running and displaying a rundown.
2. In the Configuration Tool, open App Configuration and confirm the required setup is complete, including the MOS communication settings that let Showmaker act as a newsroom system towards Viz Mosart.
3. Consider the scenarios under *Optional Server Setups* below.
4. Continue with Getting Started.

Studios

A *Studio* groups the MOS devices that make up one production environment, typically a main Viz Mosart server with an optional backup, or one gallery on a multi-studio site. Showmaker uses studios to decide which Viz Mosart server a rundown is sent to, and operators can override the studio on an individual rundown.

Studios are created in the MOS communication settings, and a site running several Viz Mosart back ends from one Mosart Web Apps host defines them on the **Studios** page. See Studios.

Show Settings

The settings that decide how a whole show behaves live on a **Show settings** page of their own, rather than in the **Edit show** dialog. Open it from the settings glyph in the top bar of the landing page, the rundown page, the rundown template page, an open library story or the Show Assets page, tooltip **Show settings (P)**, or press **P**.

The page is administrator only. The button is not shown to a user without the `showmaker-admin` role, and a non-admin who follows a direct link is returned to the show library.

There is no **Save** and no **Cancel**. Each setting is committed on its own shortly after you change it, and the top bar reports the state: **Saving…**, then **All changes saved**, which fades. If a save cannot go through you see **Couldn't save changes**, **Fix the highlighted errors to save** while a table row is invalid, or **Choose a template to save** while auto-select has no template picked. Leaving the page commits whatever is still outstanding, keeping the rows that are valid and reverting the ones that are not.

Leave the page with the back arrow, shortcut **B**, which returns you to wherever you opened it from, or the home icon, shortcut **H**, for the show library. If the rundown or template you came from has been deleted in the meantime, the arrow says so instead of making the trip, and offers **Return to library**.

What remains in **Edit show**: the show title, its description, **Starts at** and **Ends at**, the banner image and the studio.

Default option for new rundowns

DEFAULT OPTION FOR NEW RUNDOWNS decides what new rundowns in this show start from: **Blank rundown**, or **Auto-select template** together with a **Set default template** dropdown. Auto-select stays unavailable until the show has at least one template, with the tooltip **Create a rundown template first**, and the page offers a link to the Templates tab.

Copyright © 2026 Vizrt

Reading-speed presets

Turn on **Auto-calculate planned duration from script** and the **Reading-speed presets** table appears. Switching the feature on for the first time seeds one usable preset, so script timing works immediately.

Each row is a **Preset name**, typically the presenter's name, and a **Reading speed** in words per minute. A radio button in the row marks the show's default.

- A show may hold at most **four** presets. **Add preset** disappears once the fourth exists.
- A name is required and must be unique within the show.
- The speed must be a whole number between **60** and **400** wpm.
- The default preset cannot be deleted; set another preset as default first. The last remaining preset cannot be deleted while the feature is on. The bin is disabled in both cases and its tooltip gives the reason.

Deleting a preset asks for confirmation, because it is not a neutral edit. The dialog names the preset and spells out the consequence: story items using it fall back to the show's default preset, and any durations calculated from script are recalculated. On a show that has somehow lost its default, it says instead that those planned durations stay as they are until a default is set.

Switching the feature off discards any preset edits made while it was on and restores the saved rows.

Story duration

AUTO-CALCULATE STORY DURATION sets each story's planned duration from the sum of its item durations. Under it, **Which items count toward the total?** offers **All primary items** or **Only selected types**. Choosing selected types reveals a checkbox for each primary item type and pre-ticks them all.

Story properties

A show can define its own vocabulary for its stories: a named property and the list of values a story may choose from. Journalists then pick a value per story from a dropdown instead of inventing their own wording. See Working with Showmaker for how a story uses them.

In the **STORY PROPERTIES** section, the count beside the heading is the number of complete definitions. Each row is a **Property** name and its **Values (comma-separated)**, typed as one comma-separated list, for example `Ready, In progress, Needs approval`. **Add property** adds a row, **Add first property** appears while the show has none, and the bin icon removes one.

The page enforces these rules:

- A definition needs a name once it has values, and at least one value once it has a name. A row missing either is a draft: it stays on screen but is never saved.
- Names must be unique within the show, ignoring case.
- Repeated values in one list are collapsed to one.
- While any row is invalid, this section's saves are held and the top bar reads **Fix the highlighted errors to save**. The other sections carry on saving.

There is no limit on how many properties a show may define, nor on how many values a property may offer.

Giving a value a colour

A show-defined value can carry a colour, so a story's state reads at a glance in the story list. Per-story custom properties cannot, because the show has no vocabulary to colour them from.

Copyright © 2026 Vizrt

A row at rest shows its values as chips rather than as text. Click a chip and a palette opens, headed **Color · <value>**, offering ten named colours: Red, Orange, Yellow, Teal, Mint, Indigo, Blue, Purple, Pink and Black. **No color** at the foot of the palette clears it. The palette is fixed and named, so a value cannot be given a shade that is illegible on a story row.

To go back to typing the values, use **Edit values as text** at the end of the chip row, which restores the comma-separated field. It appears only once the property has at least one value, so the first value is always typed into the field.

A colour belongs to the value's text. Change a value's spelling and it becomes a value the show has not seen before, so it starts with no colour. Remove a value and its colour goes with it.

When a property or a value is removed

Nothing is rewritten on the stories. A story keeps whatever it had chosen, but the choice stops being displayed anywhere, on the story row and in the properties panel alike, and the story list can no longer be filtered by it.

Put a value back into the definition, spelled the same way, and the stories that had it show it again. Removing the whole property is different: a property added again later is a new definition, so the old choices remain stored but are never displayed again. Renaming a value has the same effect as removing it and adding another, because a value is identified by its text.

Names shared with custom story properties

Stories can also carry their own free-form custom properties, so the two kinds can collide on a name. Showmaker flags both sides, and treats the two directions differently on purpose.

On this page, defining a property whose name the show's stories already use is allowed and shows the warning **Stories already use this property name**. Taking over a name journalists already type is how such a name gets promoted to the show, so it does not block the save. Be aware of the effect: every story still using its own version of that name carries two properties under one name until the custom rows are cleared.

On the story side, a custom row that takes a name the show already defines is an error, **Already defined by the show**. The row still saves, because the properties panel has no save button, so the message is advisory and the fix is to rename or remove the row. The check reads the show's definitions live, so adding a property to the show later flags the custom rows that were already using its name.

Matching ignores case and surrounding spaces.

Story properties reach Viz Mosart

Showmaker sends a story's properties, show-defined and custom alike, to Viz Mosart with the story, as story-level external metadata. Each property arrives as a field named `sm_` followed by the property name, for example `sm_Category`. Characters not allowed in a field name are encoded, so a property named `show prop` arrives as `sm_show_x0020_prop`. Only values the story shows as chips are sent, and a story with no properties sends no extra metadata. If a story has a value for a show-defined property and a custom property of the same name, case included, only the show-defined value is sent.

To use a property in Viz Mosart, map its `sm_` name under **Newsroom Settings > Story External Metadata**, like any other story metadata field. This needs no Viz Mosart version beyond the Prerequisites above. When a property is renamed or removed or its values change, the show's stories are resent, so the fields follow the definition. A rundown that moves to another show, for example when its show is deleted and the rundowns moved, is resent too.

Copyright © 2026 Vizrt

Automatically delete old rundowns

AUTOMATICALLY DELETE OLD RUNDOWNS clears a show's old rundowns on a schedule, without anyone present. It is off for every show, and a show that never enables it is untouched.

> ⚠ **Warning**: This deletes production data and cannot be undone. There is no recycle bin. Use **Preview** to see exactly which rundowns qualify before you leave the page.

Switching it on reveals:

- **Delete rundowns older than** a number of **days**, 30 by default. The server sets the permitted range and rejects anything outside it, so the field's own messages name your site's own limits rather than fixed product ones.
- **Frequency**, either **Daily** or **Weekly**.
- **Day of week**, for a weekly schedule. It is pre-filled when you switch to weekly, so the schedule is never left incomplete.
- **Time of day**.

The schedule is stored in UTC and shown on each viewer's own clock, so the same show reads as a different local time in another region and both are the same instant. It does not follow daylight saving: a slot set for 03:00 local drifts to 04:00 local across a transition.

A rundown's age is measured from **when it was last edited**, not from its air date. A rundown carrying no modification stamp falls back to when it was created. The page does not say this, so it is worth knowing before you choose a window.

What is deleted, and what is spared

Deleting a rundown takes everything attached to it: its stories and items, its comments and its change history. Viz Mosart is told to drop it as well.

Regardless of age, the purge never touches:

- rundown templates,
- a rundown that is on air at that moment,
- a rundown whose planned start is still in the future, so a show assembled well in advance is never purged before it airs,
- rundowns in any other show. The purge is scoped to the show that configured it.

Each rundown is re-checked in the instant before it is deleted, so one edited or taken on air since the pass began is skipped.

Preview before you commit

Next to the warning, **Preview** opens **Rundowns that would be deleted**. It lists the rundowns that qualify for the retention window currently on screen, oldest first, each with its air date, or **No air date**, and when it was last edited. It reports how many would be deleted, or **No rundowns currently qualify.** At most the 100 oldest are listed, noted as **Showing the {n} oldest.**

The preview runs exactly the same selection as the purge, so it cannot promise one set and the purge delete another.

Copyright © 2026 Vizrt

When it runs

A background service checks once a minute and purges a show once its scheduled slot has passed. A missed slot, whether because the server was down or a pass was cut short, is caught up on the next check rather than waiting a whole period. After the API starts, the first pass is held back for two minutes so that the MOS connections are up, otherwise Viz Mosart would keep a rundown that Showmaker had deleted.

The first run does not fire immediately. Switching the purge on, or moving the schedule, records the moment as if a pass had just completed, so the first deletion waits for the next scheduled slot. Changing only the retention window deliberately does not, because that changes what qualifies rather than when the purge acts. That is what **Preview** is for.

At most 200 deletions are performed per minute across all shows, so a large first run drains over several minutes rather than in one burst. A show the budget did not reach stays due and continues on the next check.

Operator limits

If you need a more cautious floor than a show administrator can choose, set it on the server. These are not in the Configuration Tool: add them to the `Showmaker` section of the Web Apps server's `appsettings.json`.

Setting	Default	Effect
`Showmaker:AutoPurge:MinimumOlderThanDays`	1	The shortest retention window a show may set.
`Showmaker:AutoPurge:MaximumOlderThanDays`	3650	The longest retention window a show may set.
`Showmaker:AutoPurge:MaxDeletesPerPass`	200	Deletions performed per one-minute pass, across all shows.
`Showmaker:AutoPurge:StartupDelay`	2 minutes	How long after the API starts the first pass is held back. Capped at one hour.

There is no setting that disables the purge globally; turn it off per show. Raising the floor above a window some show already has makes the purge skip that show and log a warning once, and the show stays due until either the window or the floor is corrected.

Every pass logs what it deleted and why, naming the rundown, the show, the last-modified stamp it was judged on, the cutoff and the retention window. That log is what answers "why is this show deleting nothing?"

Roles and Access

Showmaker has two roles, which are only enforced when authentication is enabled. With authentication off, everyone has full control.

Copyright © 2026 Vizrt

Role name	Permissions
showmaker-viewer	View only.
showmaker-admin	Full control.

Roles are assigned in your authentication provider, not in the Configuration Tool, and are read from the `roles` claim. See Authentication.

Only `showmaker-admin` can open the **Show settings** page, configure the automatic deletion of old rundowns, or run its preview. The server refuses all three for a viewer, the settings button is not shown, and a direct link to the settings page returns to the show library. The keyboard shortcut list inside Showmaker, shortcut **K**, marks the administrator-only shortcuts, including **P** for Show settings.

Warning: Role checks are enforced only when both authentication and HTTPS are on. With either of them off, which is the default desktop configuration, every permission check passes for anyone who can reach the server. That includes switching on the automatic deletion of a show's rundowns. Enable authentication on any installation where that matters.

Info: A user who signs in with neither role is shown a screen explaining that they do not have access, rather than an empty show list. If that appears unexpectedly, the user is authenticated but has not been granted a Showmaker role in your identity provider. If instead the screen reports that their access could not be checked, the check itself failed and it offers a retry.

Comments and mentions

The **@mention** picker is populated from the people who have signed in to Mosart Web Apps. With authentication switched off there are no identities to harvest, so the picker stays empty and mentions cannot be used. If your teams rely on mentions, enable authentication.

Video Clips

Showmaker's clip picker searches the media service on the Viz Mosart server, so:

- **Media Administrator must be running** on the Viz Mosart server. It performs the search.
- Current Viz Mosart servers answer clip searches over the Mosart server's own real-time connection, needing no separate key. Older versions fall back to the REST media search, which requires the **Authorization Key** to be set. See Server Configuration.
- How clips are labelled in the lists is controlled by the **Video Clips** setting. See Server Configuration.

TriCaster

Two Showmaker features use a TriCaster, and both need its address and credentials entered once under **TRICASTER FOR LIVE PREVIEW**. See Server Configuration.

- **Uploading a clip** from Showmaker to the TriCaster clip server.

Copyright © 2026 Vizrt

- **Clip thumbnails** from a TriCaster, which the Mosart web server fetches on the browser's behalf so that operators need no direct access to the device and thumbnails work over HTTPS. If TriCaster clip thumbnails are blank, check the address, username and password.

To work with the Story Recorder feature

Story Recorder lets operators mark stories for recording in Viz Mosart, optionally publish the resulting clips to online platforms (TriCaster only), or replace the original story script with the recorded version. To make these controls available in Showmaker:

1. Configure Story Recorder in Viz Mosart first. See the Vizrt Documentation Center, under *Viz Mosart Administrator Guide* > *Operational Examples* > *Story Recorder Setup*.
2. In the Configuration Tool, open **Showmaker > Feature settings** and turn on **Enable Story Recorder controls**.

Once enabled, every story in Showmaker gains **Record** and **Publish** actions and a **Use recording** toggle. For how operators use these controls, see Recording a Story.

Background on Story Recorder itself is in the Vizrt Documentation Center, under *Viz Mosart User Guide* > *Operation* > *Story Recorder*.

Graphics Integrations

Showmaker can host graphics panels alongside the Viz Mosart templates. Both are optional and both are registered once in the Configuration Tool, under the Showmaker asset integration settings:

- **Viz Pilot Edge**, for editing Pilot Edge graphic templates in place.
- **Viz Flowics**, for cloud-based interactive HTML5 graphics. This needs an integration token issued by Flowics support. The field masks what you type, and the token is stored in the Web Apps server's configuration.

See App Configuration. Both panels are embedded in the browser and exchange MOS messages with Showmaker directly, so beyond the URL and MOS plugin ID there is no further server-side setup.

9.6.3 Optional Server Setups

These settings are usually performed by a system administrator.

Support for main and backup connection

- If you have a *backup server* for redundancy, define the environment on the Studios page, where each studio carries its own main server and optional backup.

Support for HTTPS

- If you are using *HTTPS*, define the environment in HTTPS and Certificates.

9.6.4 Troubleshooting

See section Troubleshooting.

Copyright © 2026 Vizrt

10 Smart Panel

- Why Smart Panel
- Where Smart Panel Fits
    - Optional integrations
- Key Features
- Typical Workflow

Smart Panel is a web-based tool for building and operating custom panels that drive Viz Mosart. It is aimed at operators, technical directors, producers and presenters who want a tailored control surface for their show: the buttons, faders, timers and previews they need, arranged the way they want, on any device with a browser.

10.1 Why Smart Panel

Live control rooms have always relied on hardware panels: dedicated boxes of buttons and faders wired to the automation system. They work, but every change to the show means a hardware change, every new operator means a new physical workstation, and every studio ends up with a different layout that lives only in someone's head.

Smart Panel takes the opposite approach. Panels are designed in the browser and opened on any screen: a touch monitor in the gallery, a tablet at the presenter's desk, a laptop in a remote control room. An operator can drop in a take button, a story countdown, a live preview and an audio fader, arrange them, and have a working panel in minutes.

Smart Panel is also designed to feel like the operator's own surface. Each panel has its own name, aspect ratio and visual identity, so the same Viz Mosart system can host a presenter panel, a director panel and an audio panel side by side without confusion. The aim is a control surface that is quick to build, easy to share, and pleasant to use on air.

> **Info**: Smart Panel does not replace Viz Mosart itself. It is a configurable front-end for Viz Mosart's automation, timing and audio commands.

10.2 Where Smart Panel Fits

Smart Panel is one of the **Viz Mosart Web Apps**, a suite of browser tools served by a single backend that ships with every Viz Mosart installation. The other apps in the suite include the Rundown Viewer, Showmaker, the NRCS Plugin, the Prompter, the Named Overlays Editor, the Template Editor and the Configuration Tool. They share a host and a single sign-in, so once you reach one of them you can move between the others without signing in again.

Within the suite, Smart Panel plays the role a hardware control panel would play in a traditional setup: it sends automation commands to Viz Mosart and shows live state back to the operator.

Component	Role for Smart Panel	Required?
Viz Mosart Server	Receives Smart Panel's commands (take, cue, transition, audio, and so on) and streams back live timing, rundown and device state.	Yes

Copyright © 2026 Vizrt

Component	Role for Smart Panel	Required?
Mosart Web Apps Configuration Tool	The one-off setup that points Smart Panel at the Viz Mosart server or servers, and registers the optional integrations described below. See Setup and Administration of Smart Panel.	Yes (once)
Mosart Web Apps server	Hosts all the web apps (sign-in, navigation, shared help), stores Smart Panel panels and uploaded images, and does the work for features the browser cannot do alone, such as receiving NDI video.	Yes (bundled)
Mosart Web Proxy	The reverse proxy that routes browser traffic to the right backend, either the web apps or the Viz Mosart server.	Yes (bundled or external)

Smart Panel uses the Mosart Web Apps server for storage, while live data and command execution go to the Viz Mosart server. If Viz Mosart is offline the panel still opens, showing placeholder values, so panels can be designed without a live show on air.

10.2.1 Optional integrations

Some panel features reach beyond Viz Mosart. Each is registered once in the Configuration Tool and then becomes available to every panel on the system. Set up only the ones you need.

- **TriCaster**: makes the studio's TriCaster inputs and outputs available as live preview sources, and lets a button take a TriCaster macro.
- **Media Sequencer**: lets a button take an MSE graphic, and lets an asset list read its content from a Media Sequencer playlist or show.
- **Viz Arc**: lets a button run a Viz Arc action.
- **Pilot Data Server**: lets a Pilot Edge Action Panel widget host a Viz Pilot Edge template.

Info: NDI needs no entry in the Configuration Tool. The Mosart Web Apps server discovers NDI sources on the network itself and can also publish a finished panel back out as an NDI source. Both directions require the NDI Runtime on the Mosart Web Apps host. Earlier releases used a separate Viz Coder service for this; that is no longer the case, and Coder does not need to be installed.

10.3 Key Features

- **Panel editor**: design panels on a configurable grid, with an item palette, a properties pane, a per-panel aspect ratio, as many pages as a panel needs, and sections whose contents buttons switch. Ready-made preset panels give you a starting point instead of an empty canvas.
- **A wide item library**: buttons, a rehearsal-mode switch, timers, labels, static text and images, live video, audio faders, connection and device status, the prompter, asset lists, a graphics playlist, embedded web

Copyright © 2026 Vizrt

pages, sections, and widgets for the timeline, story script, current and next item type, audio toggles, video transition, Viz Pilot Edge templates and the live routing of a Viz Mosart Media Router.

- **Live integration with Viz Mosart**: buttons trigger automation commands, from template takes and timeline control to audio, overlays and transitions, while timers, labels and indicators update in real time.
- **Beyond Viz Mosart**: buttons can also drive a TriCaster, a Media Sequencer or Viz Arc, so one surface can cover more than the automation system.
- **Live video in and out**: show NDI, TriCaster, streamed or file-based video inside a panel, and publish the panel itself as an NDI source for a video device or mixer.
- **Multiple panels**: build separate panels for different roles, such as director, audio and presenter, and open the right one per device.
- **Browser-based and device-agnostic**: runs in any modern browser, on touch screens and standard monitors, tablets and laptops.
- **Edit and display modes**: design a panel in the editor, then open it in display mode on the operator's screen to run it on air.
- **Ready check**: a running panel says in plain words whether its buttons will work, and before the show it can get Viz Mosart ready.
- **Roles**: where authentication is enabled, panel authors and panel users can be separated, so operators can run panels without being able to change them.

10.4 Typical Workflow

1. Open Smart Panel and create a panel, either blank or from a preset, giving it a name and an aspect ratio.
2. Place items from the palette onto the grid and configure each one: its text, command, source and colors.
3. Save the panel.
4. Open it in display mode on the operator's screen and use it to drive Viz Mosart on air.

For the step-by-step version, see Get Started. For the full reference, see Working with Smart Panel.

Copyright © 2026 Vizrt

10.5 Get Started

This page walks you through opening Smart Panel, recognizing the main views, the core terminology, and building and running your first panel.

For an overview of Smart Panel and where it fits, see the Smart Panel introduction.

- Open Smart Panel
- Tour of the Interface
- Key Terms
- Your First Panel

10.5.1 Open Smart Panel

Smart Panel runs in a web browser. Open it at `http(s)://<server>:55142/smartpanel` (port `55143` for HTTPS), where `<server>` is the hostname or IP of the Mosart Web Apps host. Your administrator may have changed those ports.

You can also open Smart Panel from **Home** in the Configuration Tool. On a multi-studio site that page shows a section per studio, and Smart Panel is studio scoped, so the card launches against the studio you have selected.

Smart Panel is one of the Mosart Web Apps, so the same host serves the other apps in the suite, and switching between them does not require signing in again.

Signing in and what you are allowed to do

Whether you sign in depends on how your administrator configured Mosart Web Apps:

- **Authentication off (the default install)**: you reach Smart Panel without signing in, and everyone has full control. Each device that runs a panel is normally paired once with a code from the Configuration Tool before its command buttons work; the panel asks for the code itself. A panel link that carries the REST API key (?apikey=<key>) also still works. A site with **Manual proxy setup** keeps using such links unless its administrator has saved the key and set its own proxy to send the Smart Panel requests to Mosart Web Apps. See Setup and Administration of Smart Panel.
- **Authentication on**: the first time you open the page you are redirected to your organization's identity provider and signed in there. What you can do then depends on your role.

Role	What you can do
smartpanel-admin	Full control: create, edit, duplicate and delete panels, as well as run them.
smartpanel-viewer	Open and use panels, but not edit them. Opening a panel takes you straight to display mode.

Roles are granted in your identity provider, not in Smart Panel. If you sign in and are told you do not have access, you are authenticated but have not been given either role: ask your administrator. For the administrator's side of this, see Setup and Administration of Smart Panel.

Copyright © 2026 Vizrt

Once signed in, a Smart Panel holds its sign-in for as long as its page is open. It is not signed out for inactivity, so a panel left running on a control room or studio screen stays usable through a show without anyone touching it. An administrator can choose to let Smart Panel time out when idle instead; see Setup and Administration of Smart Panel.

If a panel does show **You will be logged out soon**, or **Session timeout!** once it has expired, either that setting has been turned on for Smart Panel, or the browser could not reach the Mosart Web Apps host for longer than the configured session length.

> **Note**: If Smart Panel opens but an **Offline** chip stays in the app bar, the connection to Viz Mosart could not be established. You can still design panels, but live values are replaced by placeholders. See the troubleshooting notes in Setup and Administration of Smart Panel.

10.5.2 Tour of the Interface

Smart Panel has three main views.

Landing page

At `/smartpanel`, this lists the saved panels under the heading **Your panels**. From here you create a new panel or open an existing one.

- Click a panel to open it. If you can edit, it opens in the editor; if you can only view, it opens in display mode.
- The **Search panels...** field above the cards filters the list by name, which is what you reach for once a site has more panels than fit on a screen.
- Each card has a **Copy link** button, for handing the panel to whoever needs to run it.
- The card's options menu offers **Display mode**, **Duplicate Panel** and **Delete Panel**.
- **New** opens the **Create new panel** dialog, where you start from a **Blank panel** or from one of the ready-made panels under **Use a preset**. The button is shown only to a user who can edit panels.

Panel editor

At `/smartpanel/<panel>/edit`, this is where you design a panel. It has four areas:

1. **App bar** (top): a back arrow, the panel name (click to rename), an **Edit** and **Preview** switch so you can try the panel without leaving the editor, a **Copy link** button and a play button that opens the panel in display mode in a new tab, **Save**, an **NDI Stream** control for publishing the panel as an NDI source, a more-options menu holding **Duplicate Panel**, **Copy Link** and **Delete Panel**, and the **?** help icon. An **Offline** chip appears here when Viz Mosart is unreachable.
2. **Components palette** (left): the fifteen item types you can place on the canvas, namely Timer, Label, Text, Image, Widget, Web page, Prompter, Asset list, Graphics, Status, Fader, Button, Rehearsal, Live and Section. Below it, in the same column, the **Layers** pane lists everything already on the panel. Drag the divider between them to give **Layers** more room.
3. **Canvas** (center): the panel grid. Click a palette item, then click an empty cell to drop it. Click an existing item to select it. Above the canvas, the page bar has a tab for each page of the panel and a **+** button to add one.
4. **Properties pane** (right): headed **Properties**, with two tabs. **Component** holds the settings of the selected item, and **Panel** holds the settings of the whole panel. Selecting an item switches the pane to **Component**, and clearing the selection switches it back to **Panel**.

Copyright © 2026 Vizrt

Tip: Both side panes can be resized by dragging their inner edge, and pressing **K** shows the keyboard shortcuts.

Display mode

At `/smartpanel/<panel>`, without `/edit`, the panel runs touch-friendly and full-width, ready for the operator's screen. This is what you use on air, and it cannot be edited. Reach it from the **Display mode** action on the panel card, or from a copied link. For a completely bare screen, use the browser's own full-screen mode.

Help menu

The **?** icon at the top right of every page opens the help menu:

Option	Opens
Keyboard shortcuts	The shortcut reference. Also available by pressing **K**.
Documentation	This Smart Panel user documentation.
What's new	The release notes for the current version.
Give feedback	An email to the Smart Panel team (`mosart-feedback@vizrt.com`).
About	Smart Panel version and build information.

User avatar. When you are signed in, your avatar appears at the top right next to the help icon. Click it for your account and to sign out. Smart Panel shares its identity provider with the other Mosart Web Apps, so signing in or out applies to them all in the same browser session.

10.5.3 Key Terms

Term	Meaning
Panel	A single panel design: a grid of items on one or more pages, plus its own name, aspect ratio, background, padding and text size presets. Stored on the server, so everyone who can reach it sees the same panel.
Page	One screen of a panel, with its own items. The pages of a panel share its grid and settings. In display mode the operator changes page with page buttons on the panel.
Item	One thing on a panel: a button, timer, label, image, fader, and so on.

Copyright © 2026 Vizrt

Term	Meaning
Edit mode	The panel open at `/smartpanel/<panel>/edit`, with the app bar, the palette and the properties pane.
Display mode	The panel open at `/smartpanel/<panel>`, with no `/edit`. This is what you run on air, and it cannot be edited.
Grid	The rows and columns the canvas is divided into. Items snap to cells. Set on the **Panel** tab.
Aspect ratio	The shape of the canvas. Choose a preset, set a custom ratio, or pick *None* to stretch to fit the window. Set per panel on the **Panel** tab, so each panel suits its target screen.
Panel tab	The **Panel** tab of the **Properties** pane, opened automatically when no item is selected. Holds the groups **Panel** (the fill color), **Grid settings** (rows, columns and aspect ratio), **Page navigation** on a panel with several pages, **Component presets** (the default padding and the text sizes that items refer to as S, M and L) and **Ready check**.
Component tab	The **Component** tab of the **Properties** pane, opened automatically when an item is selected. Its header carries the item's name, which you double-click to rename, an eye button that shows the name on the item, and the **Component actions** menu with **Copy**, **Duplicate** and **Delete**, plus entries that move the item to another page or into or out of a section when the panel has them. Below come **Position and size**, the settings of the item's type, any **Interaction**, and **Appearance**.
Layers	The searchable list of every item on the panel, in the left column under the palette. Useful when items overlap or the panel is busy, and drag a row by its handle to change the stacking order.
Offline	A chip in the app bar meaning the connection to Viz Mosart is not established. Live data is replaced with placeholder values so you can keep designing.
Preset panel	A read-only starting point offered under **Use a preset** in the **Create new panel** dialog. What you create from one is a normal, editable panel.

Copyright © 2026 Vizrt

10.5.4 Your First Panel

A short walk-through that uses every core concept once. It assumes Smart Panel is connected to a working Viz Mosart system. If the **Offline** chip is showing you can still complete the design steps, and only the last step needs a live server.

1. **Create a panel.** On the landing page, click **New**, choose **Blank panel**, name it *My first panel* and click **Create**. Smart Panel opens it in the editor.
2. **Set the canvas shape.** With nothing selected, the right pane opens on its **Panel** tab. Pick an **Aspect ratio** that matches your target screen, for example a landscape ratio for a wall-mounted touch monitor. The same choice is offered in the **Create new panel** dialog when you start from a blank panel. Leave the grid as it is for now.
3. **Add a take button.** Click **Button** in the palette. The cursor picks up a preview. Click an empty cell near the bottom of the canvas to drop it, and its settings open on the **Component** tab on the right. Set the **Label** to *Take*. Under **Interaction**, click **Add interaction**, leave the **Source** as **Mosart**, set the **Command** to **Template**, choose **Program** as the **Bus**, then pick any template from the **Template** picker, for example a camera template.
4. **Add a story countdown.** Click **Timer** in the palette, then click a cell at the top of the canvas. On the **Component** tab, choose **Current Item Countdown** from the **Variant** drop-down. The item shows a placeholder countdown, or a live one if Viz Mosart is connected.
5. **Add a label.** Click **Label**, click a cell next to the timer, and choose **Current Item Slug** from the **Variant** drop-down so it shows the name of the on-air item. Set **Text size** to *M*.
6. **Try it without leaving the editor.** Switch the app bar from **Edit** to **Preview**. The palette and handles step out of the way so you can see the panel as an operator would.
7. **Save.** Click **Save** in the app bar. The panel is stored on the server, and anyone who can reach it can now open it.
8. **Run it.** Go back to the landing page and choose **Display mode** on your panel's card, or use **Copy Link** and open that link on the operator's screen. Tap the take button and, with Viz Mosart connected, it sends the template you bound in step 3. On a site without OIDC sign-in, a device that has not been paired yet first shows a **Pair this device** card instead: type the **Pairing code** from the Configuration Tool and a name for the device, and the button works. See Setup and Administration of Smart Panel.

You have now built and operated your first panel. To go deeper into any step, see Working with Smart Panel.

Copyright © 2026 Vizrt

10.6 Working with Smart Panel

This section is the working reference for everyday use of Smart Panel: managing your panels, shaping the canvas, placing and configuring panel items, binding buttons to commands, and using a finished panel on air. It assumes you have followed Get Started at least once and have a working connection to Viz Mosart.

- Managing Panels
- The Editor
    - The app bar
    - Panel properties
    - Working with Items
    - Pages
- Item Reference
    - Timer
    - Label
    - Text
    - Image
    - Widget
    - Fader
    - Button
    - Rehearsal
    - Live
    - Section
- Running a Panel On Air
    - Ready check
    - Streaming a panel as an NDI source
    - What to avoid while live
    - Touch operation
- Keyboard Shortcuts

10.6.1 Managing Panels

All panels live on the Mosart Web Apps server, so anyone who can reach the Smart Panel URL sees the same list and works on the same panels. The **landing page** (`/smartpanel`) is the entry point for panel management. It is headed **Your panels**.

Smart Panel also ships with a small library of read-only **preset panels**. Presets are not listed on the landing page and cannot be edited in place: they are offered as starting points under **Use a preset** in the **Create new panel** dialog. Anything you create from one is a normal, editable panel, and the preset itself stays untouched.

From the landing page you can:

- **Create a panel.** Click **New** to open the **Create new panel** dialog. Choose **Blank panel** to start from an empty canvas, or one of the panels under **Use a preset** to begin with a ready-made layout. Give it a **Name** and click **Create**; Smart Panel opens it in the editor. The **New** button is shown only to a user who can edit panels.
- **Find a panel.** The **Search panels...** field above the cards filters the list by name, which is what you reach for once a site has more panels than fit on a screen.

Copyright © 2026 Vizrt

- **Open a panel.** Click its card. If you have edit rights it opens in the editor at `/smartpanel/<panel>/edit`; if you can only view panels it opens straight in display mode.
- **Open a panel to run it.** Choose **Display mode** from the card's options menu, or navigate to `/smartpanel/<panel>` without `/edit`.
- **Hand a panel to an operator.** Use the **Copy link** button on the card. The link opens the panel in display mode. It carries no key, so it can be shared; on a site without OIDC sign-in, a device that has not been paired asks for a pairing code the first time.
- **Duplicate a panel.** Choose **Duplicate Panel** from the card's options menu, or from the more-options menu inside the editor.
- **Rename a panel.** Open it in the editor and click the panel name in the app bar.
- **Delete a panel.** Choose **Delete Panel** from the card's options menu, or from the editor's more-options menu. Deletion is permanent and removes the panel for everyone, so confirm before deleting a panel that may be in use.

Info: Panel names do not have to be unique. Two panels may share a name, so if your team relies on names to tell panels apart, agree on a convention.

Note: If you try to leave the editor with unsaved work, Smart Panel asks what to do rather than discarding it silently. Save before you navigate away if you want to keep the changes.

10.6.2 The Editor

The editor has an app bar across the top and three working areas below it:

- The **Components** palette on the left, holding the item types you can place.
- The **canvas** in the middle, which is the panel itself, with the page bar above it: a tab for each page of the panel, and on the right the panel's aspect ratio and the number of components on the page shown.
- The **Layers** list on the left under the palette, and the **Properties** pane on the right, which has a **Component** tab for the selected item and a **Panel** tab for the whole panel.

Both side panes can be resized by dragging their inner edge, so you can give the canvas more room while working on a dense panel, or widen the properties pane when configuring an item with many settings.

The app bar

The app bar carries the panel name, which you click to rename, and:

- An **Edit** and **Preview** switch. **Preview** shows the panel as an operator would see it without leaving the editor, which is the quickest way to check spacing and legibility. A user who can only view panels never sees this switch: the editor is not open to them, and an edit link takes them to display mode instead.
- **Save**, which is available only while the panel has unsaved changes. There is no **Discard** button in the app bar. If you try to leave the editor with unsaved work, a **Save changes** dialog offers **Save**, **Discard** and **Cancel**.
- **NDI Stream**, which publishes this panel as an NDI source and stops it again.

Copyright © 2026 Vizrt

- Two icon buttons next to the connection chip: **Copy link**, which copies the display-mode link to the clipboard, and a play button, which opens this panel in display mode in a new tab. The play button is the quickest way to see the panel exactly as the operator will.
- A more-options menu holding **Duplicate Panel**, **Copy Link** and **Delete Panel**, with **Audio Countdowns** added at the top only on a panel that uses audio. **Duplicate Panel** is unavailable while the panel has unsaved changes, and says so in a tooltip; **Delete Panel** is unavailable on a read-only preset.
- The **?** help menu, and an **Offline** chip whenever Viz Mosart is unreachable.

Info: the panel actions menu, click-to-rename, **Save** and **NDI Stream** all appear only for a user who can edit panels. A viewer does not reach the editor at all: opening a panel, or following an edit link, takes them to display mode, which has no app bar.

Tip: Press **K** at any time for the keyboard shortcut list. See *Keyboard Shortcuts* at the end of this page.

Panel properties

With no item selected, the right pane opens on its **Panel** tab. These settings apply to the canvas and govern the whole panel, in the groups **Panel**, **Grid settings**, **Page navigation** (on a panel with several pages), **Component presets** and **Ready check**. Selecting an item switches the pane to the **Component** tab, and you can open **Panel** again at any time.

Property	What it controls
Fill (**Panel** group)	The color behind all items. Pick something that contrasts with your item colors and is comfortable to look at for a long show.
Rows and Columns (**Grid settings**)	How finely the canvas is divided. Items snap to cells. Increase the counts for fine-grained panels, decrease them for chunky touch buttons.
Aspect ratio (**Grid settings**)	The shape of the canvas. Pick one of the presets, choose *Custom* and enter your own width and height, or choose *None* to stretch to fit the browser window. Match this to your target screen so the panel looks right on air.
Page navigation	The page buttons of a panel with several pages. See *Pages* below.
Padding (**Component presets**)	The padding of every item that does not set its own.
Small (S), Medium (M) and Large (L) (**Component presets**)	Three named text sizes, defined once here in pixels. Items then refer to them as **S**, **M** or **L** instead of carrying their own pixel values. Changing a preset updates every item that uses it.

Copyright © 2026 Vizrt

Property	What it controls
Ready check	What the running panel says about whether its components will work, and what its Ready sheet may do before the show. See *Ready check* under *Running a Panel On Air*.

Working with Items

Panel items are the building blocks of a panel. They all behave the same way on the canvas, whatever their type.

Adding an item. Drag an item type from the palette onto the panel, or click the item type and then click where you want it. Either way, a label with the item type's icon and name follows the pointer until you reach the canvas. Over the canvas, the cells the new item will take are outlined instead, centred on the pointer as far as the grid allows, and inside the section under it where there is one. When you click first, the item waits until you place it. To cancel, press **Escape**, click the item type again, or click outside the canvas and the palette. Once placed, its settings appear in the **Properties** pane.

Selecting an item. Click it on the canvas, or its entry in the **Layers** list. The selected item shows its handles and its properties populate the pane.

Moving and resizing. Drag the item body to move it, and a corner or edge handle to resize it. Items snap to the grid, and while you drag, the cells the item will snap to are outlined beneath it. With an item selected you can also nudge it with the **arrow keys** and resize it with **Shift** and the arrow keys.

Copying an item. Use **Ctrl+C** and **Ctrl+V** (**Cmd** on a Mac), or the **Duplicate** action in the **Layers** list. To drop a copy where you want it, drag the item and hold **Alt** (**Option** on a Mac) as you let go: the copy lands there and the original stays where it was.

Deleting an item. Select it and press **Delete** or **Backspace**, use the **Delete** action in the **Layers** list, or drag it onto **Drop to delete**, which appears at the bottom of the canvas while you move an item. The message that follows offers **Undo**.

The Layers list. Every placed item appears in this searchable list, in the left column under the palette. Use it when items overlap and you need to reach one underneath another, when the panel is busy and you want to jump straight to something by name, or simply to confirm what is on the panel. Drag a row by its handle to change the stacking order. A section's row names the variant it shows, and lists that variant's components under it, which the row can fold away. Each row has an options menu with **Copy**, **Duplicate**, **Bring to front**, **Send to back**, the moves to other pages and into or out of sections, and **Delete**, and the list header has one with **Copy all**, **Paste** and **Clear all**.

> **Tip**: Give each item a descriptive name by double-clicking its name at the top of the **Component** tab (for example *Take button*, *Story countdown*). The name appears in **Layers**, and naming as you go pays off quickly as the panel grows.

Item properties

The **Component** tab is the same for every type from top to bottom: a header, **Position and size**, a group named after the item's type with the settings only that type has, and **Appearance**. A Button, a Live item and a web page add an **Interaction** group for what a press does.

- **The header** carries the item's name, which is the name used in **Layers**. Double-click it, or use the **Rename** button, to rename the item; **Enter** keeps the name and **Escape** cancels it, and an emptied name goes back to the automatic one. The eye button decides whether the name is also drawn on the item itself, and the

Copyright © 2026 Vizrt

Component actions menu holds **Copy**, **Duplicate**, the moves to other pages and into or out of sections, and **Delete**.

- **Position and size**: **X** and **Y**, the column and row of the item's top left cell counted from 0, and **W** and **H**, its width and height in cells. Inside a section they count the section's own grid. The fields follow the canvas while you drag, and a number typed in moves the item at once.
- **Appearance** on a Timer, Label, Text or Button: **Fill** and **Text** colors, entered as hex values or picked from the swatch, which opens a palette with **No fill (transparent)** where it applies and **Custom color** for any other color; **Bold font**, **ON** or **OFF**; **Align**, left, center or right, which a Button does not offer; and **Text size**, **S**, **M** or **L** from the panel presets or **Fill** to scale the text to the item bounds, between **Min text** and **Max text**.
- **Padding**, at the end of **Appearance** on every type, defaulting to the panel-wide value. An Image adds a **Fill** color of its own, and the other types have nothing else there.

While a setting such as **Padding**, **Min text**, **Max text** or a color holds a value of its own, its name is underlined: click the name to put the default back.

Pages

A panel can hold several pages, so one screen can carry more controls than fit on it at once, for example the cameras on one page and the audio on another. The pages share the panel's grid, aspect ratio, background and presets, and each page has components of its own. A new panel starts with one page, *Page 1*.

The page bar above the canvas has a numbered tab for each page:

- **Add a page** with the **+** button after the tabs. The new page is empty, is named *Page 2*, *Page 3* and so on, and opens at once. A panel can have up to 100 pages.
- **Switch page** by clicking its tab. With a tab focused, the arrow keys, **Home** and **End** move between pages. Switching page clears the selection.
- **Rename a page** by double-clicking its tab, or by pressing **F2** on a focused tab. **Enter** keeps the new name and **Escape** cancels it.
- **Delete a page** with the close button on its tab, or by pressing **Delete** on a focused tab. The **Delete page** dialog says how many components go with the page, and warns when components elsewhere on the panel go to it when pressed. The last page cannot be deleted.
- **Move a component to another page** with **Move to** and the page's name, in the component's actions menu or in its row menu in the **Layers** list. It keeps its position where those cells are free on the other page, and takes the first free spot otherwise.

On the landing page a panel with several pages says how many it has, and its card shows the first page, which is the page the panel opens on.

Page buttons

Display mode has no page bar, so the operator changes page with page buttons on the panel itself. Adding a second page turns them on, and deleting the panel back to one page turns them off. They are set in the **Page navigation** group of the **Panel** tab, which appears once the panel has more than one page:

- **Page buttons**: **SHOW** or **HIDE**.
- **Layout**: **Single** puts one button in a corner, chosen under **Corner**, which steps to the next page and from the last page back to the first. **Prev + next** puts a previous and a next button in the two corners of one side, chosen under **Side** as the top or the bottom.

Copyright © 2026 Vizrt

Each page button takes one grid cell and shows the page on screen out of the total, for example *2/3*. It sits on top of any component in that cell, so keep the cell free. In the editor the buttons only mark their cells and do not react; in display mode, and in the editor's **Preview**, they switch page. With the page buttons hidden, a Button with the **Navigate to page** command can switch pages instead.

10.6.3 Item Reference

The palette holds **fifteen** item types, three to a row.

Item	What it is for
Timer	A live timer or countdown driven by Viz Mosart. The basis of a timing display for the control room or the studio floor.
Label	A live text value from Viz Mosart, such as the current item slug or story name.
Text	Static text that does not change at runtime. Titles, group headers, instructions, branding.
Image	A static image uploaded to the server. Logos, role icons, decoration.
Widget	A category hosting eight built-in variants, from the timeline to a Viz Pilot Edge template and the Media Router's routing matrix.
Web page	An external web page embedded in the panel, with an optional click action.
Prompter	The Prompter's rolling script.
Asset list	A filtered list of the assets in the rundown, each a take.
Graphics	A Graphics Playlist of the rundown's overlay graphics, to preview, take in and take out.
Status	The Viz Mosart connection state, or the state of each Viz Mosart device.
Fader	An audio fader bound to a Viz Mosart audio channel.
Button	The main interactive item. Sends a command when tapped, and can carry a second action on the next tap.
Rehearsal	A control that shows whether Viz Mosart is in rehearsal mode, and flips it when tapped.

Copyright © 2026 Vizrt

Item	What it is for
Live	A live video preview, with an optional click action.
Section	An area of the page that shows one of several sets of components at a time, switched by buttons.

Timer

Shows a live timer or countdown driven by Viz Mosart, synchronized with the active rundown. Use it to build timing displays: one on the studio floor counting the current item, another in the control room counting packages to keep the producer updated.

Pick the value from the **Variant** drop-down in the **Timer** group. There are twenty-one:

Group	Types
Current item	Current Item Countdown (the default), Current Item Time
Current story	Current Story Countdown, Current Story Planned Duration, Current Story Time
Next	Next Break Time, Next Item Duration, Next Live Countdown, Next Story Planned Duration
Rundown	Current Rundown Out, Rundown Elapsed Time, Rundown Over/Under, Rundown Out Over/Under
Time and break	Time of Day, Time to Next Break, Time to Rundown Out, Duration to Next Break, Duration to Rundown Out
Story Recorder	Story Recorder Preroll Countdown, Story Recorder Recording Time, Story Recorder Countdown

Besides the shared properties, a Timer offers **Overtime accent**, **SHOW** or **HIDE**. Shown, the timer changes its background color once it passes zero, which is what you want for countdowns and over/under timers. The same setting is offered on a Label, so a label that reports a negative value can carry the same background change.

> **Info**: There is no per-threshold coloring. Over-time appearance comes from **Overtime accent** plus the **Fill** color you choose, not from separate warning and critical colors.

Clock values

Two of the timer types show a wall clock rather than a duration: **Time of Day** and **Next Break Time**. Both follow the time zone of the machine the panel is open on, so a panel on a studio screen in another region reads the local time there rather than the automation system's. They are written according to that browser's language settings, always with a two-digit hour.

Copyright © 2026 Vizrt

An administrator can change both of those choices for the whole installation, selecting the Viz Mosart server's time zone instead, or setting the format explicitly with a 12 or 24 hour clock and a regional format. The controls are on the **Smart Panel** tab of **App Configuration** in the Configuration Tool; see App Configuration. A panel reads them when it loads, so reload a panel that is already open after they change.

Note: This needs Viz Mosart 5.17 or later. Against an older server the time zone setting has no effect on **Time of Day**: it shows the Mosart server's wall clock whichever zone is selected, which is what to suspect when that clock reads a time zone you did not choose. The format setting still applies to it. **Next Break Time** does follow the time zone setting against an older server, but with the Mosart server time zone selected it arrives already formatted by the server, so that is the one value that can show AM and PM on a 24 hour site.

Label

Shows a live text value from Viz Mosart. The same role as a Timer, but for text. Pick the value from the **Variant** drop-down in the **Label** group; there are eleven:

Group	Sources
Current	Current Item Slug (the default), Current Story Name, Current Video Server Channel, Current Graphics Channel
Next	Next Item Slug, Next Story Name, Next Video Server Channel, Next Graphics Channel
Rundown and server	Rundown Name, Server Description
Panel	Page Title, the name of the page the label is on. It comes from the panel itself, so it shows whether or not Viz Mosart is connected.

Info: The list is fixed. There is no free-form expression, so a Label cannot be bound to an arbitrary Viz Mosart value that is not in the list. Long values can overflow a narrow item, so either size it generously, drop to a smaller preset, or set **Text size** to **Fill** with sensible bounds.

Text

Static text, typed by you rather than read from Viz Mosart. Use it for section titles such as *Cameras* or *Audio*, group headers, on-screen instructions, and branding such as *Studio 3*.

The only type-specific property is **Custom text**, the text itself. Manual line breaks are preserved, so you can lay out a caption over several lines instead of relying on wrapping. The same applies to button labels.

Info: Text items never update on air. If the value should change with the show, use a Label instead.

Copyright © 2026 Vizrt

Image

A static image uploaded to the Mosart Web Apps server and stored with the panel, so every device that opens the panel gets the same file. Use it for studio logos, small role icons next to related buttons, and visual markers that help an operator orient themselves.

- **Image**: its upload button opens a file picker, and once a file is uploaded it reads **Replace**. Allowed formats are **.jpg**, **.png**, **.gif** and **.webp**, and the maximum file size is **5 MB**. Until something is uploaded, the item shows a placeholder on the canvas.
- **Scaling** has five values: **Fill** covers the whole item and may cut off the picture's edges, **Fit** shows the whole picture inside the item and may leave empty space, **Stretch** fills the item exactly and may distort the picture, **Original size** shows the picture at its own size, centred, and **Shrink to fit** behaves like Fit but never enlarges. A new image starts on **Fill**, 4 cells wide and 2 high.
- **Fill**, in the **Appearance** group, is the color that shows through wherever the scaling leaves space, or where the image itself is transparent.

> **Info**: There is no opacity property. To dim an image for a watermark effect, bake the transparency into a PNG before uploading it.

Widget

Widget is a category rather than a single item. Drop one on the canvas and pick which of its **eight** built-in variants it shows from the **Variant** drop-down in its **Widget** group. A variant with settings of its own adds a group named after it below.

Variant	What it shows
Timeline	The Viz Mosart timeline for the current story.
Story Script	The script for the current story.
Current Item Type	A visual indicator of the on-air item's type, such as camera, package or graphic.
Next Item Type	The same, for the next item.
Audio Toggles	Read-only indicators for five Viz Mosart audio modes: HA for hold audio transition, K for keep sound levels, FM for fade manual, L2 for use level 2 preview and an underlined L2 for use level 2 on air. Each lights up when its mode is on. **Indicators** picks which of the five appear. They report state only, so tapping one does nothing.

Copyright © 2026 Vizrt

Variant	What it shows
Video Transition	Read-only indicators for the video transition: HV for hold video transition, M for mix, W for wipe, E for effect, and a fifth cell carrying the transition rate in frames, or the effect name when an effect is set. **Indicators** picks which of the five appear.
Pilot Edge Action Panel	A Viz Pilot Edge template hosted in action mode, as a permanent control surface. See below.
Media Router Matrix	The live routing of a Viz Mosart Media Router, as a matrix of inports and outports. See below.

Five more live views have tiles of their own in the palette, and their settings sit in a group named after the tile:

Component	What it shows
Status	**Shows** picks one of two views. **Connection**: the Viz Mosart connection state, shown as a beacon for the main server, with a second beacon for the backup where redundancy is configured. Hovering a beacon reports the Viz Mosart Server version it is connected to, and while the main server is idle, tapping its beacon on a running panel offers to make it active, confirmed with a hold, where the panel's Ready check allows it. Otherwise the panel says to make it active in Viz Mosart. **Devices**: live status indicators for the connected Viz Mosart devices, showing connected, disconnected or standby.
Asset list	A filtered list of the assets in the rundown. See below.
Graphics	A Graphics Playlist: the rundown's overlay graphics as a table, story by story, to preview, take in and take out. See below.
Prompter	The Prompter's rolling script, embedded from the Prompter app in its display-only mode. The Prompter's controls, status strip and banners are hidden, so the component shows the script and a thin progress bar at the bottom of the frame and nothing operable. To start and stop the roll from the panel, add a Button with the **Prompter Play/Pause** command.
Web page	An external web page embedded in the panel. See below.

Copyright © 2026 Vizrt

Apart from Web page, Pilot Edge Action Panel and Media Router Matrix, all of these are driven by Viz Mosart and need a live connection to show real data. Without one they show placeholder content.

Asset list

The most configurable component. It lists the assets available in the show and can be narrowed down heavily, so several asset lists on one panel can each cover a different job, for example manual overlays, overlays on the current story, fullscreen graphics, and everything in the rundown.

- **Asset source**: **Mosart rundown**, or **Media Sequencer** for a playlist or show. Choosing Media Sequencer requires the Media Sequencer to be configured, and replaces the groups below with **Source**, holding the **MSE profile**, the **Source type** (playlist or show) and the **Source** to read from, and **Display**, holding **Thumbnail** and **Group name**.
- **In the widget**: **Story filter** and **Asset filter**, **SHOW** or **HIDE**, decide whether the operator gets those two filters in display mode, so they can narrow the list themselves. The choice under each applies whether it is shown or not, as where the list starts or as what it always lists: **Stories** is All, On Air or Next, and **Assets** is All, Scheduled or Manual.
- **Display**: under **Sort by**, Timeline keeps the rundown order and Template groups the items by template type. Under **Layout**, List gives one row per asset, and Tiles wraps the assets as tiles in a card per story, sized with **Tile size** (S, M, L or XL). **Thumbnail** shows or hides thumbnails, with their **Size**, and **Story name** and **Script** add context to each row. Tiles does not offer **Script**.
- **Stories to include**: **Story slug filter**, **Show the stories whose type is** and **Show the stories whose status is**: narrow the list story by story before any asset is looked at, so a story that fails is left out with all its assets. The story types are Camera, Package, Voice-over, Live, Graphics, DVE, Telephone and Float, and a story of none of these types, such as a segment break, always stays. The story statuses are Valid, On air, Next, Aired and Other: the story on air answers to On air alone and the next story to Next alone, whatever status they carry, and a status none of the boxes names counts as Other.
- **Assets to include**: the filters below, applied to the assets of the stories that remain.
- **Show the assets whose type is**: show or hide each of Camera, Video, Live, Graphics, DVE, Float, Overlay graphic and Accessories independently.
- **Destination filter**: restrict graphics to the graphics destinations you list, each written in full. An asset with no destination always stays.
- **Slug filter** and **Template variant filter**: show only the items whose slug or template variant matches one of the terms. Put * at either end of a term as a wildcard, for example *goal* or CAM*. The story slug filter takes wildcards the same way. For an overlay graphic with a graphics destination, the Template variant filter matches the destination instead of the template variant.
- **Show the assets whose status is**: show or hide Valid, Invalid, On air, Aired and Other items independently. The item on air answers to On air alone and an item that has aired to Aired alone, whatever status they carry.
- **Show the assets whose template is**: Valid for items with a template that is present and not empty, Unavailable for the rest. Overlay graphics, sound items and accessories carry no template of their own and always count as Valid. Clear Unavailable to leave out items with no usable template.

The four text filters, for story slugs, destinations, slugs and template variants, each hold a list of terms. Type a term and press **Enter**, or click the **+** button beside the field, to add it as a chip. Click a chip's text to edit it, and its close button to remove it. A comma-separated list pasted into the field becomes one chip per term. An item passes when any term matches, case is ignored, and a filter with no terms lets everything through.

Copyright © 2026 Vizrt

Note: Filtering by story type and story status needs Viz Mosart 5.19 or later. Against an older server the story type boxes have no effect, and every story other than the one on air and the next one counts as Other.

Rows carry the item slug and, for overlay graphics, their on-air status, so an operator can see what is already live before taking something. Items the Viz Mosart GUI added to a story itself, such as a direct take, are not listed, and an asset that appears twice in a story is listed once. In display mode, and in the editor's **Preview**, a row is also a take: tapping it takes that asset, a template for a rundown asset and the graphic for a Media Sequencer source. As with a Button there is no confirmation, so place an asset list where an operator will not brush it under pressure.

Graphics Playlist

The **Graphics** tile adds a Graphics Playlist: a table of the rundown's overlay graphics, story by story, from which an operator previews graphics, takes them in and takes them out. Each row shows the graphic, its destination and how it leaves air: after a duration, at the end of its item (*Item end*) or story (*Story end*), or only when taken out (*Open end*). The story on air is highlighted, a graphic on air carries an **On air** badge, and graphics that have aired are dimmed. A **Graphics Playlist** preset panel holds one, ready to use.

In display mode, and in the editor's **Preview**, each row has **Preview**, which sends the graphic to the preview engine, and **Take in**. On a graphic that is on air, the **On air** badge is its **Take out** button. A graphic with continue points plays in steps: taking it in shows its first state, and each continue moves it on to the next. When any graphic in the playlist has continue points, the table gets a **Continue** column (with **Row buttons** on). Off air it shows how many points a graphic has, such as *3 points* (*1 point* for one). On air with points left it is a button that names the point it takes next, such as **Continue 2/3**, or just **Continue** while the panel does not yet know how many points have been taken. Once the last point is taken, the column shows the progress, such as *3/3*, and no button. Each continue sweeps across its row, including a continue made from the Viz Mosart GUI or another panel. In the editor the rows show live data but the buttons do nothing, and the widget says so. Once the playlist has focus it also works from the keyboard: the arrow keys move between graphics, **Page Up** and **Page Down** move between stories, **Space** sends the graphic to preview, **Enter** takes it in and, on a graphic on air with continue points left, continues it, **Delete** or **Backspace** takes the graphic out if it is on air, **Home** takes the last graphic out and **Ctrl+X** takes all graphics out.

Its settings are grouped as follows:

- **In the widget**: **Destination filter** shows a filter in the Destination column header, with which the operator turns destinations on and off. **Shortcuts button** shows a button at the left of the header that lists the playlist's keys. **Row buttons** shows **Preview** and **Take in** on each graphic, and a **Continue** column when any graphic has continue points; hidden, the table shows a status column instead, and a graphic with continue points shows a small mark beside its name: the number of points or, on air once known, how many have been taken out of the total.
- **Display**: **Thumbnail** and its size, **Story names** for the page number and story name on each story's first graphic, and **Empty stories** to list stories with no overlay graphics too, so the table follows the whole rundown.
- **Graphics to include**: **Destinations**, a list of destinations, each written in full, saved with the panel. With none, every destination is listed, and the operator's filter can narrow the list further.

Note: Previewing, taking in, taking out and continuing from the Graphics Playlist needs Viz Mosart 5.19 or later.

Copyright © 2026 Vizrt

Pilot Edge Action Panel

Hosts one Viz Pilot Edge template in its action mode, so an operator fires the template's own controls from the panel. You pick the template by title from your Viz Pilot Data Server, and a template designed for action mode works as it is.

The Pilot Data Server is set once for the whole installation, on the **Server Configuration** page of the Configuration Tool. If none is set there, Smart Panel uses the server of a Viz Pilot Edge plugin configured for Showmaker or the Named Overlays Editor, unless that plugin is on localhost. Without either, you cannot pick a template, and the widget says so: *No Pilot Data Server is configured. Set it up on the Server Configuration page of the Web Applications Configuration Tool.* See Server Configuration.

- **Select template ...**: opens a searchable list of the templates on the Pilot Data Server. Once one is chosen, its title and ID are shown, with a pencil to change it.
- **Edit in Template Builder**: opens the chosen template in a new tab, in the Template Builder set on the Server Configuration page.
- **Show template name**: adds a header carrying the template title above the frame.
- **Set a playout target**: when on, the template uses the Media Sequencer configured for the Web Apps as its playout target, with the **Profile** and **Channel** you set here. You pick them from lists read from that Media Sequencer, or type them when it cannot be read. Turn the switch off if the template's script sets its own target. With the switch on and no Media Sequencer configured, or only localhost, the frame stays empty and the widget says what is missing.

In the editor the frame is inert, so clicks select the widget and nothing fires. In display mode, and in the editor's **Preview**, the template is live. Signing in to a secured Pilot system happens in the operator's browser, with the method chosen in the Configuration Tool.

> **Info**: None of the preset panels uses this variant, because it needs your site's Pilot Data Server.

Media Router Matrix

Shows the routing of a Viz Mosart Media Router live, as a matrix: inports as rows and outports as columns, each grouped by category, such as Video or Vision Mixer, and colored by it. A filled dot marks each crosspoint routed now. When a take moves a route, the new crosspoint blinks and the one it replaced fades out. Where an inport and an outport of different categories meet, the crosspoint is dimmed. The widget only shows the routing: nothing in it takes a route. It needs Viz Mosart 5.19 or later, the first version whose Media Router has the SignalR service the widget connects to.

Above the matrix, the router's name carries a beacon: green while the router is active, orange while it is idle (a slave, which takes no routing commands), and gray while it is not answering. Hovering the name reports the router's address and, once it answers, its version. Hovering a port name reports its device and whether that device is connected, its category and its status, among other details. Hovering a routed crosspoint names its inport and outport, and a requested one adds *Pending: not confirmed by the client yet*.

Its settings, in the **Media Router Matrix** group:

- **Media Router address**: the address of the Media Router's SignalR service, its host and the HTTP port set in MediaRouterServiceConfig.xml, 55171 unless set otherwise, such as `http://mmr-host:55171`. For an `https://` address, use the HTTPS port set in the same file instead. Include the port. If you leave out the scheme, the address is taken as `http://`. An entry that is not a web address is flagged when you leave the field: *Not a web address. Enter one such as http://mmr-host:55171.* On a panel served over HTTPS, an

Copyright © 2026 Vizrt

`http://` address is flagged too: *This page uses HTTPS, so the browser blocks an http address. Use the router's https address.*

- **Pending**: **SHOW** or **HIDE**. Shown, a crosspoint requested and not yet confirmed by the client device is drawn as an outline. Shown by default.
- **Unused ports**: **SHOW** or **HIDE**. Hidden, the matrix keeps only the inports and outports routed now, and those of the latest change, both the new and the previous one, so a sparse matrix stays small. When no port is left to show, the widget says *Nothing is routed.* Shown by default.

Problems show as a message on the panel, once for each router rather than at every retry:

- *Set the Media Router address in the widget options.*: the address is empty.
- *The Media Router address is not a web address.*: correct the address in the widget options.
- *This page uses HTTPS, so the browser blocks the Media Router at <host>, which uses HTTP. Turn on HTTPS in the Media Router's SignalR service and use its https address.*
- *Cannot reach the Media Router at <host>. Retrying…* or *Lost the Media Router at <host>. Reconnecting…*: the widget keeps trying, and the matrix it last had stays on screen, dimmed, until the router answers again.
- *The Media Router <name> is idle (slave): it takes no routing commands.*
- *The Media Router <name> has no ports to show.*

In the editor the matrix shows live data but is inert, so clicks select the widget. The messages about the address do not appear there: an entry that is not a web address, or an `http://` address on a panel served over HTTPS, is flagged under the **Media Router address** field instead, and an empty address shows no message at all.

Info: None of the preset panels uses this variant, because it needs your site's Media Router address.

Web page

- **URL**: the address to embed, an `http://` or `https://` address. If you leave out the scheme, an address such as `example.com/news` is taken as `https://`. This works when the host has a dot or is localhost. A one-word host, such as `grafana:3000`, needs its `http://` or `https://` typed. The frame is loaded directly. An entry that is not a web address is flagged when you leave the field.
- **Interaction**: **Add interaction** makes the whole frame act as a button, with the same options a Button offers under **On press**. Without an interaction, clicks pass through to the embedded site.

Note: The embedded site decides whether it may be framed. If it sends `X-Frame-Options: DENY` or a restrictive `Content-Security-Policy: frame-ancestors`, the frame stays blank and Smart Panel cannot override it. The editor checks this and warns under **URL** that the site does not allow other pages to show it.

Fader

An audio fader bound to a Viz Mosart audio channel. The operator drags the bar to change the level on air. It renders as a vertical bar with the channel name at the top, the current level in dB below the name, and the track and its scale beneath them, showing `−∞` at the floor.

Copyright © 2026 Vizrt

Use it for any source that needs continuous control: a presenter microphone, a music bed, a remote feed. The **Channel** drop-down is populated from the audio channels Viz Mosart is currently reporting, so name each fader after its source (INPUT 1, INPUT 2 and so on) and operators can tell at a glance which channel they are riding.

> **Note**: The channel list only exists while Viz Mosart is reachable. With the connection down the drop-down is empty, so faders can only be bound while Viz Mosart is connected.

Button

The main interactive item. Its **Button** group holds **Variant**, **Single state** or **Two-state**, and the **Label** drawn on it. A new button does nothing until you click **Add interaction** in its **Interaction** group. Under **On press** the button then picks *what it talks to*, its **Source**, and then *what it sends*. The bin button in the **Interaction** title removes the interaction, and the message that follows offers **Undo**.

Choosing the source

Source	What you can send
Mosart	A template take, a timeline command, a control command, Prompter play/pause, or one of the panel's own page and section commands. See below.
TriCaster	A TriCaster macro, chosen by folder and then by name.
MSE	Take a graphic. Pick the Media Sequencer profile, whether the graphic comes from a playlist or a show, and then the element itself.
Viz Arc	A Viz Arc action, chosen by project and then by action.

The TriCaster, MSE and Viz Arc sources need their integration configured first. See Setup and Administration of Smart Panel.

Viz Mosart commands

With **Mosart** as the **Source**, the **Command** is one of:

- **Template**: takes a Viz Mosart template. Choose the **Bus**, **Program** to take on air or **Preview** to cue, then pick the **Template** from a picker that lists the whole template catalogue by category and lets you search it.
- **Timeline**: one of ten timeline actions, namely Toggle Rehearsal Mode, Open Rundown, Reload, Start Continue, Skip Next, Unskip Next, Skip Next Sub Item, Unskip Next Sub Item, Start From Top, and Take Server Out Of Idle. **Open Rundown** can point at a specific rundown, or leave the choice to the operator at runtime.
- **Control Command**: the full Viz Mosart control command set, from autotake and direct take through overlay graphics, audio, crosspoints and studio setup. Commands that need arguments take them in the parameters

Copyright © 2026 Vizrt

field, in URL query form (`param1=value1&param2=value2`). For example `engine_switcher` needs an engine name. Leave it blank for commands that take none.

- **Prompter Play/Pause**: starts and stops prompter scrolling.
- **Switch section variant**: changes which variant a section shows, without sending anything to Viz Mosart. Pick the **Section**, then the **Variant**: **Next variant** or **Previous variant**, which wrap round, or one of its variants by name. It is offered once the page has a section, and a button inside a section switches its own section unless you pick another. A button that sits in only one variant of the section it switches hides itself when it switches away, and the editor says so: put a way back in every variant, or keep the button on the page beside the section.
- **Navigate to page**: switches the panel to another page, without sending anything to Viz Mosart. **Go to** offers **Next page**, **Previous page** or one of the pages by name. It is offered once the panel has more than one page. If the page a button goes to is deleted, the button names it **Deleted page**, and a press says the page no longer exists until you point it at another one.

Test, in the title of **On press**, sends the configured command straight from the editor, so you can confirm a binding without leaving it.

> **Note**: **Test** sends the real command to the connected Viz Mosart server. Do not use it during a live show unless you mean to. Control command parameters are also passed through as typed: a typo fails silently from the operator's point of view, so check the Viz Mosart server log if a button appears to do nothing.

Two-state buttons

A button can carry a second action. Set **Variant** to **Two-state**: **Label** becomes **State 1 label**, **State 2 label** is added, shown while the button is armed for its second action, and the **Interaction** group splits into **On first press** and **On second press**, each with its own **Test**. The first tap sends the first action, the next tap sends the second, and the button then resets to the first state. Both halves take the full range of targets and commands, so one cell can hold a take and its matching take out.

If you leave **State 2 label** empty, the button keeps its ordinary label in both states, so give it a second label whenever the operator needs to see which action the next tap will send.

Switching sections as well

Besides its action, a button can switch one or more sections, on its own page or on another, so that one press both sends a command and shows the controls that go with it. Under the action, **And switch a section** adds an **And switch** row, where you pick the **Section** and the **Variant** it should show: **Next variant**, **Previous variant** or one of its variants by name. A button can switch up to 10 sections, and each row has a button that removes it. A section the button's own **Switch section variant** command already switches is not offered. A button can also switch sections without any action of its own. On a two-state button each press has its own rows.

The sections switch only once the action has gone through, or straight away when there is no action, so the panel never shows a state that did not happen: if the command fails, nothing switches. A button whose command lets the operator choose the rundown when the button is pressed only opens the rundown picker: its sections do not switch, not even after a rundown is picked. A two-state button likewise moves on to its other state only when the press went through. As with **Switch section variant**, a row that switches the button's own section away from the only variant the button is in hides the button, and the editor says so. If a section a row points at is deleted, the row shows **Deleted section**, and a press says that a section it switches no longer exists.

A running panel does not confirm the switch, because the operator sees the sections change. When you test the button in the editor, a message says which sections it switched and what they now show. If it switches more than

Copyright © 2026 Vizrt

three, the message counts the sections instead of naming them, and names what they show only when they all show the same variant.

Appearance

Buttons carry the full color and font controls, so use color and a short, unambiguous label (*TAKE*, *CUE*, *CAM 1*) to make the purpose obvious at a glance.

> **Note**: A button fires as soon as it is tapped. Smart Panel does not ask the operator to confirm a button press, so for an action that must not go to air by accident, rely on placement, size and colour, and keep it away from the buttons an operator reaches for under pressure. A two-state button is not a safety guard either: its first tap fires immediately.

Rehearsal

A control with two states that *reflects* the current state in Viz Mosart as well as changing it, so the operator can see at a glance which way it is set and does not have to look elsewhere to confirm it. That reading back of the state is what separates it from a Button: a plain Button can also have two states, but it never shows you what Viz Mosart currently thinks.

The state it flips is Viz Mosart's **rehearsal mode**. Placed on a panel, it lights up while Viz Mosart is in rehearsal and switches it when tapped. Its **Rehearsal** group holds only the **Label** drawn on it, *Rehearsal* unless you type another.

> **Info**: If you want a one-way action instead, or another state, use a Button. **Toggle Rehearsal Mode** is also available as a plain timeline command on a Button, which flips rehearsal without showing its state.

Live

Embeds a live video preview on the panel. Use it for camera and source previews, and any other picture the operator needs to glance at. A common arrangement is a Live preview paired with its own take button, both labelled with the same source name so the link between what you see and what you cut to is obvious.

Source type offers four choices:

Source type	What to select
NDI	An NDI source, picked from the sources Mosart Web Apps has discovered on the network. The server receives the NDI stream and passes it to the browser, so nothing needs installing on the operator's machine.
TriCaster	A TriCaster input or output, from the TriCaster configured in the Configuration Tool.
HTML Video	A video URL the browser can play directly, such as an `.mp4` or `.webm` file. Useful for pre-recorded clips.

Copyright © 2026 Vizrt

Source type	What to select
Stream	A `ws://` or `wss://` OvenMediaEngine signalling URL, or an `http://` or `https://` stream or video URL.

Each source type remembers its own URL, so switching between them while experimenting does not lose what you entered for the others.

Add interaction, in the **Interaction** group, turns the preview into a button as well: the same options a Button offers appear under **On press**, so tapping the picture can take that source. Without an interaction, a press on the preview does nothing.

> **Info**: If a source cannot be reached, the picker says which part failed rather than simply showing an empty list. The messages and their remedies are listed under Setup and Administration of Smart Panel. Earlier releases required a separate Viz Coder service for NDI; that is no longer used.

Section

A section is an area of the page that holds several sets of components, its **variants**, and shows one of them at a time. Buttons on the panel switch which variant it shows, so one area of the screen can serve several purposes in turn.

A new section covers 8 by 6 cells and has a grid of its own of the same size, so its cells line up with the page's until you change either. It starts with one variant, *Variant 1*. Sections are named *Section 1*, *Section 2* and so on, in reading order on their page, unless you give one a name of its own.

Putting components in a section. Drop a component from the palette onto the section, or drag a component on the page into it: it goes into the variant the section shows in the editor. Drag it out past the section's edge to put it back on the page. Hold **Ctrl** (**Cmd** on a Mac) while dragging to keep a component on the grid it is on: a page component can then sit on top of a section, and a component in a section snaps back inside it. The component's actions menu and its row menu in the **Layers** list also offer **Move into** a section and **Move out of section**. A section always sits on the page, never inside another section.

A section's own properties are:

- **Fill**: its background color, which can also be transparent.
- **Variants**: the list of variants, each with the number of components in it. The selected variant is the one the canvas shows and you edit. The **+** button adds a variant, up to 50. Double-click a variant, press **F2** on it or use its rename button to rename it, and its delete button to delete it. Deleting a variant that holds components, or that a button switches to, asks first. The last variant cannot be deleted.
- **Section grid**: the section's own **Rows** and **Columns**, and **Default component padding** for the components inside it, which follows the panel's default until you set it.

In the editor a section draws its grid and, once it has more than one variant, names the variant it shows. Deleting a section that holds components, or that a button switches, asks first. A running panel opens every section on its first variant, and the panel's buttons switch it from there; see **Switch section variant** under *Viz Mosart commands*. The editor's **Preview** runs the panel, so its buttons switch sections there too, and back in **Edit** each section shows the variant it showed before.

Copyright © 2026 Vizrt

10.6.4 Running a Panel On Air

To check a panel while you are still designing it, switch the app bar from **Edit** to **Preview**, or use the play button in the app bar to open it in display mode in a new tab.

To run it for real, open it in display mode: choose **Display mode** on the panel's card, use **Copy Link** and open that link on the operator's screen, or navigate to `/smartpanel/<panel>` without `/edit`. The panel fills the window, touch-friendly and ready for use, and cannot be edited there. For a completely bare screen, use the browser's own full-screen mode.

On air:

- Timers, labels and indicators update live from Viz Mosart.
- A panel with several pages opens on its first page. The page buttons, and any button with the **Navigate to page** command, switch page.
- Buttons send their command as soon as they are tapped. A two-state button sends its second action on the next tap, then resets. A button that also switches sections switches them once its command has gone through.
- Faders apply level changes as they are dragged, but only while the Viz Mosart audio mixer is connected. Otherwise the fader is dimmed and its handle will not move, and tapping the handle reports why: the audio mixer is not configured on the Mosart Server, is in standby, or is not connected.
- Display mode has no app bar, so there is no **Offline** chip there. Instead the Ready check says on the panel itself when the connection drops or Viz Mosart cannot act, unless it is turned off; see *Ready check* below. A **Status** component set to **Connection** also shows the server state on the panel.

Where authentication is enabled, a panel holds its sign-in for as long as its page is open, so running a panel on air does not put a sign-in dialog over the controls: the session does not expire under the operator. An administrator can choose to let Smart Panel time out when idle instead, in which case a warning appears before it does; see Setup and Administration of Smart Panel.

Ready check

A running panel checks whether its components will work and says so in plain words, so whoever stands at the panel knows before a press fails. How much it shows is set in the **Ready check** group of the **Panel** tab, under **Visualize**:

- **Off**: nothing is shown. Use it for clocks and monitors that are in shot.
- **Strip only**: a status strip while something is wrong. The default for a panel with no components that send Viz Mosart commands.
- **Strip and sheet**: the strip, plus a Ready sheet before the show that can get Viz Mosart ready where you allow it. The default for a panel with components that send Viz Mosart commands. A panel with no such components shows the strip only.

A panel follows these defaults until you change something in the **Ready check** group, and only what you change is saved. So a panel with components that send Viz Mosart commands, where nothing has been changed, shows the strip and the sheet with every action of the sheet switched on. This includes panels made in an earlier version. To keep a panel quieter, choose **Strip only** or **Off**, or switch off the actions it should not offer.

The status strip runs along the top of the panel and names the problem and what it affects, for example *No connection*, *Can't reach Mosart*, *Mosart is not running*, *Mosart is idle* or *No rundown loaded*. It never takes a tap: a press always reaches the component under it. While the show runs, the strip gives way to a small badge in a corner

Copyright © 2026 Vizrt

the components leave free. If every corner is taken, it goes where it covers the least. While Viz Mosart cannot act on them, components that send it commands are dimmed and carry a mark, and a press still sends. A time that stops updating gets a *Not updating* mark, and after a few more seconds shows placeholders.

The Ready sheet covers the panel only before the show, while nothing is on air, and closes when the show starts. It offers what you allow in the **Ready check** group. With **Strip and sheet**, on a panel with components that send Viz Mosart commands, all three switches are on by default:

- **Panel can make Mosart active**: a *Mosart is idle* card with **Hold to make Mosart active**. Going active sets up every studio device and can take graphics off air, and with a rundown loaded Viz Mosart also cuts to its first item. It is not offered with a main and backup server. On a **Strip only** panel the same hold is reached by tapping the beacon of a **Status** component set to **Connection** on the panel. With **Strip only** this switch starts off, so turn it on to offer the hold there.
- **Panel can load a rundown**: a *No rundown loaded* card with **Choose a rundown**, which lists the rundowns Viz Mosart offers (with a filter when there are more than 12), and loads the one chosen with **Load**. Viz Mosart checks the load, and it never replaces a loaded rundown.
- **Quick Show template** and **Panel can start without a rundown**: with a template picked and the switch on, the card also offers **Start without a rundown**, which goes on air with that template after **Hold to go on air**. With a template picked and no rundown loaded, a button with the **Start Continue** timeline command asks *Go on air now?* in the same way, and starts nothing if the template is not in Viz Mosart's template set. The **Quick Show template** field shows only while **Panel can start without a rundown** is on or the panel has a button with the **Start Continue** timeline command. Until a template is picked, the sheet does not offer **Start without a rundown**.

Not now or **Show my panel** puts the sheet away, and the strip's **Get ready** button brings it back. Actions that go on air or affect the whole gallery take a one-second hold, and the sheet's buttons ignore taps for a moment after it appears, so a finger already on its way to the panel does not press them. An action counts as done when Viz Mosart reports it, not when it is sent.

The rundown and Quick Show settings need a Viz Mosart server that reports its rundown and can start a chosen template. The **Panel** tab says when the connected server cannot, and asks you to connect to Viz Mosart before you change them. While a setting cannot apply, its switch shows off, and the note under it says when the setting is on.

Streaming a panel as an NDI source

A finished panel can also be sent out as an NDI source, so it can be taken as a picture on a video device or mixer rather than only viewed in a browser. Start and stop this with **NDI Stream** in the app bar.

A stream stays on until you turn it off. If the Mosart Web Apps server restarts, or the machine it runs on reboots, the streams that were on start again by themselves once the server is back. Deleting a panel turns its stream off.

What to avoid while live

- Editing the panel an operator is currently using. Saves take effect on their next page load, unsaved editor changes are not reflected on their screen, and a misclick in the editor can fire a real command.
- Renaming or deleting a panel that is in use.

Touch operation

- Use an aspect ratio that matches the touch monitor. A mismatch leaves dead bands at the edges that operators tap by accident.

Copyright © 2026 Vizrt

- Avoid tiny items. A button smaller than a fingertip is unreliable under pressure.
- Keep clear contrast between active and idle states so the operator can see which action just fired.

10.6.5 Keyboard Shortcuts

Press **K** to see this list in the app.

Action	Shortcut
Move between panels	Arrow keys
First or last panel	**Home** / **End**
Move the selected item	Arrow keys
Resize the selected item	**Shift** + arrow keys
Duplicate while dragging	**Alt** (**Option** on a Mac), held when you let go
Delete the selected item	**Delete** or **Backspace**
Copy item	**Ctrl+C** (**Cmd+C** on a Mac)
Paste item	**Ctrl+V** (**Cmd+V** on a Mac)
Drag without entering or leaving a section	**Ctrl** (**Cmd** on a Mac), held while dragging
Cancel placement or a drag	**Escape**
Graphics Playlist: move between graphics	**Up** / **Down** arrow
Graphics Playlist: previous or next story	**Page Up** / **Page Down**
Graphics Playlist: send to preview	**Space**
Graphics Playlist: take in	**Enter**
Graphics Playlist: continue a graphic on air	**Enter**
Graphics Playlist: take the graphic out	**Delete** or **Backspace**
Graphics Playlist: take the last graphic out	**Home**
Graphics Playlist: take all graphics out	**Ctrl+X**

Copyright © 2026 Vizrt

Action	Shortcut
Show keyboard shortcuts	K

Copyright © 2026 Vizrt

10.7 Setup and Administration of Smart Panel

- Setting up Smart Panel
- Live Video Sources
- Integrations for Button Actions
- Roles and Access
- Pairing Devices
- Multiple Studios
- Using HTTPS
- Troubleshooting

Smart Panel installs together with Mosart Web Apps. It has one page of its own in the Mosart Web Apps Configuration Tool, the **Smart Panel** tab of **App Configuration**, which sets how the clock timers are written and which time zone they follow, and which devices may run a panel. For the rest it uses the shared Mosart server, authentication and HTTPS settings. Configure those once and Smart Panel is ready.

Some panel features reach further than Viz Mosart, and each of those has its own one-time entry in the Configuration Tool: a **TriCaster** for live video and macros, a **Media Sequencer** for graphics, **Viz Arc** for Viz Arc actions, and a **Viz Pilot Data Server** for Pilot Edge templates. Set up only the ones your panels use. The **Media Router Matrix** widget needs no entry there: each widget carries the address of the Media Router it shows.

10.7.1 Setting up Smart Panel

Follow these steps in order. Each one builds on the previous, so do not skip ahead.

Step 1: Check your Viz Mosart version.
Smart Panel is part of the Mosart Web Apps suite that ships with Viz Mosart, and individual panel features depend on the Mosart server supporting them. For the Viz Mosart version required by your Mosart Web Apps release, see the *Mosart Web Apps Release Notes* at the Vizrt Documentation Center.

Step 2: Make sure Viz Mosart is running.
Smart Panel talks to a live Mosart server for automation commands, timing and rundown data. Before you continue, check that:

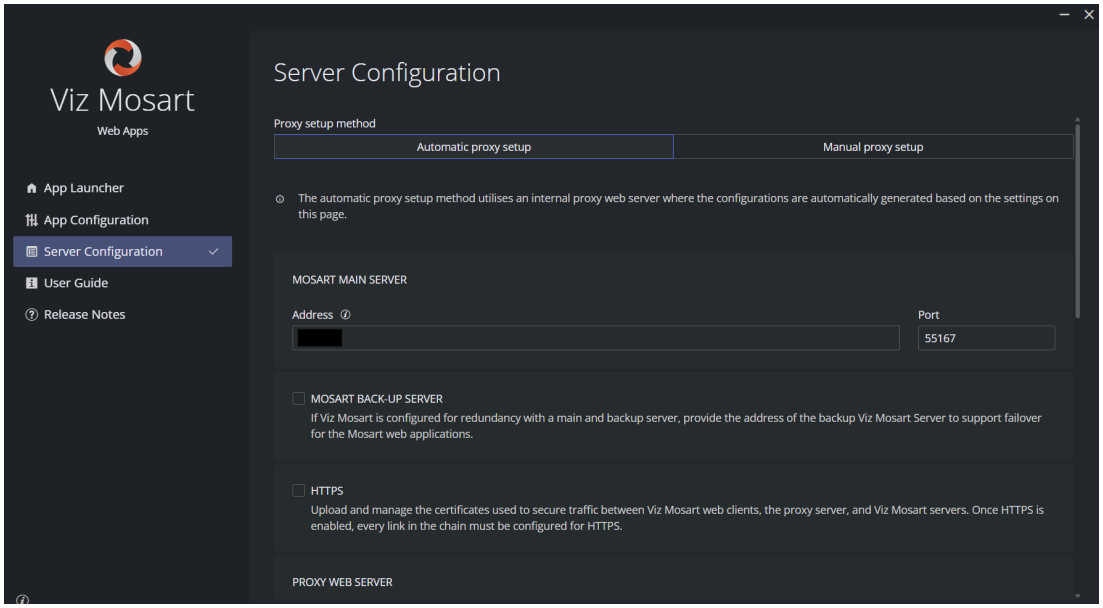
- **Manus Administrator** is running.
- The **Viz Mosart UI** is running and displaying a rundown.

If either is down, start it now. You only need to do this once after a server reboot.

Step 3: Open the Mosart Web Apps Configuration Tool.
This is where every Mosart Web App, including Smart Panel, is set up. You should already have it from the Mosart installation.

Step 4: Point the Web Apps suite at your Mosart server.
In the Configuration Tool, open **Studios** and fill in the Viz Mosart server host and port for your studio, together with its optional backup server.
Smart Panel does not have a separate connection screen, it uses these shared settings to talk to the Mosart server.

Copyright © 2026 Vizrt



Step 5: (Optional) Enable live video previews.
If you plan to show video in the **Live** element, set up the sources you need now. See *Live Video Sources* below.

Step 6: (Optional) Set up the integrations your panels need.
Buttons can drive a TriCaster, a Media Sequencer or Viz Arc as well as Viz Mosart. The Pilot Edge Action Panel needs a Viz Pilot Data Server. The Media Router Matrix connects to the Media Router's SignalR service itself, new in Viz Mosart 5.19, so that service must be turned on and reachable from operators' browsers. See *Integrations for Button Actions* below.

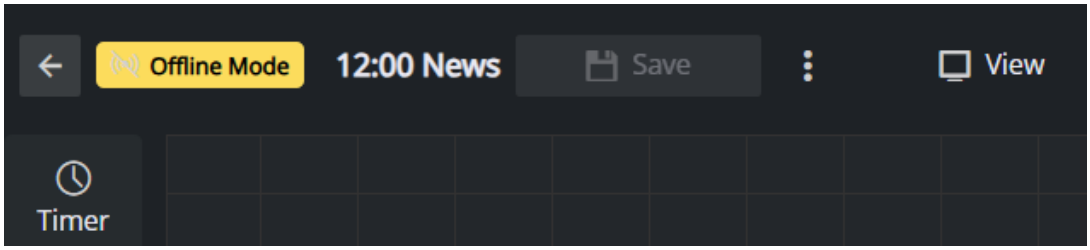
Step 7: (Optional) Turn on HTTPS and authentication.
By default Mosart Web Apps runs without HTTPS or sign-in. If your site needs them, configure both in the Configuration Tool. See *Using HTTPS* and *Roles and Access* below.

Step 8: (Optional) Set how the clock timers are written.
In the Configuration Tool, open **App Configuration** and choose the **Smart Panel** tab. Under **TIME FORMAT**, leave **Use browser default** on to follow the language settings of each Smart Panel machine, or clear it and pick a regional format, a 12-hour or 24-hour clock and whether the hour carries a leading zero. In the Configuration Tool window on the server, **Use this PC's settings** fills those in from the server's Windows regional format; in a browser, **Use this browser's format** fills them in from that browser. Under **TIME ZONE**, choose **Local (browser)**, the default, or **Mosart server**. Both settings apply only to the Time of Day and Next Break Time timers, and a panel reads them when it loads, so reload any panel that is already open.

Step 9: Open Smart Panel and verify.

Browse to `http(s)://<your-mosart-web-apps-host>:port/smartpanel` (the default port is 55142 for HTTP and 55143 for HTTPS). You should see the landing page, empty on a fresh install, headed **Your panels**, with a **New** button ready. Open or create any panel and check that the **Offline** chip in the app bar is **not** shown. If it is, the connection to Viz Mosart did not come up. See *Troubleshooting* below. On a site without OIDC sign-in, the panel then asks for a pairing code before its buttons work; see *Pairing Devices* below.

Copyright © 2026 Vizrt



Tip: Place a **Status** component on a panel and set **Shows** to **Connection** to see the Mosart server state as a small beacon, with a second beacon for the backup server where redundancy is configured. Hovering a beacon reports the Viz Mosart Server version it is connected to. While the main server is idle, tapping its beacon on a running panel offers to make it active, confirmed with a hold, where the panel's Ready check allows it. See *Ready check* in Working with Smart Panel.

10.7.2 Live Video Sources

The **Live** element supports four source types. Only the first two need anything configured:

Source type	What to configure
NDI	Nothing in the Configuration Tool. Mosart Web Apps discovers NDI sources on the network itself and streams the one you pick to the browser. The **NDI Runtime must be installed on the Mosart Web Apps host**; without it the source list reports that the NDI runtime is not available on the server.
TriCaster	The TriCaster address, username and password, entered once under **TRICASTER FOR LIVE PREVIEW**. Both TriCaster inputs and outputs are then selectable. See Server Configuration.
HTML video	Nothing. Point the element at a video URL the browser can fetch directly, for example an `.mp4` or `.webm` file. Useful for pre-recorded clips.
Stream	Nothing on the Mosart side. Enter either a `ws://` or `wss://` OvenMediaEngine signalling URL, or an `http://` or `https://` stream or video URL.

Info: Live video works over HTTPS. Both NDI and TriCaster are served by the Mosart web server on the browser's behalf, over the same origin as the rest of the app, so there is no separate host for the browser to reach and no mixed-content problem. Earlier releases required a separate Viz Coder service for NDI; that is no longer used and does not need to be installed.

Copyright © 2026 Vizrt

Streaming a panel out as NDI

A panel can also be sent the other way, as an NDI output, so a finished panel can be taken as a source on a video device or mixer. This is started and stopped from the panel's own app bar and uses the same NDI runtime on the Mosart Web Apps host, so nothing extra is configured for it.

The streams that are on are saved on the Mosart Web Apps host, in *smartPanelNdiOutputs.json* in the same folder as the panel layouts file, *smartPanelLayouts.json*, and start again when the server starts. Back up *smartPanelNdiOutputs.json* together with the layouts file.

10.7.3 Integrations for Button Actions

A button can target Viz Mosart, a TriCaster, a Media Sequencer or Viz Arc. Viz Mosart needs nothing beyond the server address from Step 4. The others each need their one-time entry in the Configuration Tool:

- **TriCaster**: uses the same TriCaster credentials as live video. With those in place, a button can take a TriCaster macro, chosen by folder and name.
- **Media Sequencer**: enter the Media Sequencer address and REST API port (default 8580) under **MEDIA SEQUENCER**. This enables buttons that take an MSE graphic, asset lists that read their content from a Media Sequencer playlist or show, and the playout target of a Pilot Edge Action Panel. The playout target needs an address other than localhost. See Server Configuration.
- **Viz Arc**: enter the address and port of the Viz Arc REST web service (default 9004) under **VIZ ARC**. The Configuration Tool probes the host as you type and tells you whether the service answers. The Viz Arc REST web service requires its own licence, and only the Mosart Web Apps host needs network access to it. See Server Configuration.

One widget has its own one-time entry too. The **Pilot Edge Action Panel** widget lists templates from a **Viz Pilot Data Server**: enter its host name and port under **PILOT DATA SERVER** (8177 for HTTP, 7373 for HTTPS). If you leave **Address** empty, Smart Panel uses the server of a Viz Pilot Edge plugin configured for Showmaker or the Named Overlays Editor, unless that plugin is on localhost. Operators' browsers connect to the Pilot Data Server directly, so it must be reachable from them, not only from the Mosart Web Apps host. See Server Configuration.

The **Media Router Matrix** widget has no entry in the Configuration Tool: its **Media Router address** is set in each widget. Operators' browsers connect to the Media Router's SignalR service directly, so it must be reachable from them, not only from the Mosart Web Apps host. That service is off by default: turn it on in the Media Router's MediaRouterServiceConfig.xml, where its HTTP port is 55171 unless set otherwise. On a panel served over HTTPS, see *Using HTTPS* below.

10.7.4 Roles and Access

Smart Panel has two roles, which are only enforced when authentication is enabled. With authentication off, everyone has full control.

Role name	Permissions
smartpanel-viewer	Open and use panels, but not edit them.
smartpanel-admin	Full control, including creating and editing panels.

Copyright © 2026 Vizrt

Roles are assigned in your authentication provider, not in the Configuration Tool, and are read from the `roles` claim. See Authentication.

> **Info**: A user who signs in with neither role is shown a screen explaining that they do not have access, rather than an empty panel list. If that appears unexpectedly, the user is authenticated but has not been granted either Smart Panel role in your identity provider.

Session timeout

Where authentication is enabled, the Configuration Tool's **Server Configuration** page carries an **Allow session timeout** group inside **Authentication (OIDC)**, with one switch per web app and a summary line reporting how many can time out.

- **Smart Panel**'s switch is *off* by default, reading **Stays signed in while open**. A panel renews its own session for as long as its page is open, so a control-room surface is never signed out mid-show.
- Turned on, it reads **Renews while in use; can time out when idle**, which suits a shared machine where an abandoned session should not stay signed in. A panel left mid-edit is saved automatically a few seconds before the session ends.
- **Session timeout** sets how long an unused session survives, in minutes, from 5 to 10080, defaulting to 30. It applies only to the apps whose switch is on.

> **Warning**: Leave the switch off for a gallery or control-room panel. Nobody touches a panel between takes, so an idle timeout would put a sign-in dialog over the controls during a show. See Authentication.

10.7.5 Pairing Devices

Where Mosart Web Apps runs without OIDC sign-in, each device that runs a panel is paired once with the installation before its buttons work. The panel link then carries no Viz Mosart key and can be shared freely, and taking one device's access away changes no key that other panels, Stream Decks or scripts use. A signed-in panel on an OIDC site needs no pairing.

Mosart Web Apps adds the Viz Mosart server's REST API key to a paired device's commands, so that key must be entered as the **Authorization Key** on the **Server Configuration** page of the Configuration Tool. See Server Configuration.

> **Note**: The **Authorization Key** field on the **Server Configuration** page is shown only with **Automatic proxy setup**. With **Manual proxy setup**, pairing works only if the key was saved while the site still used **Automatic proxy setup**, and your own proxy sends the Smart Panel command and status routes to the Mosart Web Apps server (see *Troubleshooting*). Otherwise keep opening panels with the key in the link (`?apikey=<key>`) for now.

To pair a device:

1. Open a panel on the device. A panel that is not paired says *Your panel components are locked* and shows a **Pair this device** card, also while Viz Mosart is on air, since a locked panel cannot put anything on air. If the card has been put away, the strip's **Pair this device** button brings it back.
2. In the Configuration Tool, open **App Configuration**, choose the **Smart Panel** tab and, under **PANEL ACCESS**, click **Show a pairing code**. On a site with several studios, first pick the studio the device is for. The

Copyright © 2026 Vizrt

six-digit code works for ten minutes and pairs one device. **Generate new** issues another code, which replaces the one before. **Copy code** beside the code copies it, to send in a message.
3. On the card, type the code under **Pairing code**, give the device a name under **What to call this device (required)**, and tap **Pair**. The panel's components work at once, without a reload.

A mistyped code can be tried again, but ten wrong codes cancel it, and an administrator then has to show a new one. A pairing belongs to the browser and the studio, not to one panel, so every panel of that studio opened in the same browser works afterwards. The browser keeps the pairing in its site data: a private window, or clearing the site data, asks for a code again, and on an iPad a panel saved to the Home Screen asks once more the first time it is opened from there. A panel whose Ready check is **Off** shows no pairing card, so pair the device from another panel of the same studio.

The paired devices are listed under **DEVICES THAT CAN RUN A PANEL** on the same tab, with when each was paired and last used, and on a site with several studios the studio it belongs to. **Remove** takes a device's access away: its next button press is refused, no key changes, and the device can pair again with a new code. See App Configuration.

Info: A panel link that still carries the REST API key as `?apikey=<key>` keeps working while the key is right. Pairing is the way forward, because a key in a link is exposed in browser history and logs, and cannot be withdrawn from one device alone.

10.7.6 Multiple Studios

On a site that runs several Viz Mosart back ends from one Mosart Web Apps host, Smart Panel is opened per studio, and the panel then drives the Viz Mosart server belonging to that studio. Define the studios once on the **Studios** page of the Configuration Tool, and launch the app for a studio from the Home page. See Studios.

10.7.7 Using HTTPS

If Mosart Web Apps is served over HTTPS, define the environment using the HTTPS and Certificates page of the Mosart Web Apps Configuration Tool. Smart Panel inherits the same HTTPS setup as the other Web Apps. There are no Smart Panel-specific certificates to manage. Widgets that connect from operators' browsers straight to another server need that server on HTTPS too. The **Pilot Edge Action Panel** needs the Pilot Data Server's HTTPS port (7373 by default). The **Media Router Matrix** can reach the Media Router only over HTTPS, so turn on HTTPS in the Media Router's SignalR service and use its https address in the widget, with the HTTPS port set in MediaRouterServiceConfig.xml rather than the HTTP port 55171.

10.7.8 Troubleshooting

- The "Offline" chip stays on in the app bar
- A sign-in dialog appears over a panel
- The Live element reports a problem
- Buttons do not execute
- Layout not saved between sessions

The "Offline" chip stays on in the app bar

The **Offline** chip means Smart Panel has lost contact with Viz Mosart. The panel keeps working, but timers, labels, and faders display fixed sample values instead of live data. As soon as the connection comes back, the chip disappears and live values resume automatically.

Copyright © 2026 Vizrt

If the chip does not clear after a few seconds, work through the checks below in order.

1. **Is Mosart Server configured?** Open the Mosart Web Apps Configuration Tool, go to **Studios** and confirm the host and port of the studio's Viz Mosart server are set. Save and reopen Smart Panel.
2. **Is Mosart Server running and reachable?** On the Mosart Server machine, confirm **Manus Administrator** is running and a rundown is loaded. If it is, but Smart Panel still shows Offline, the Mosart Web Apps host can not reach the server, check the network connection between the two machines.
3. **Using a custom reverse proxy?** Smart Panel always calls `/mosart/...` on the Mosart Web Apps host (SignalR hub, REST API, status endpoints). The bundled Caddy proxy forwards `/mosart/*` to Mosart Server automatically. If you replaced Caddy with your own proxy (nginx etc.), make sure it forwards `/mosart/*` to the Mosart Server. The exceptions are Smart Panel's commands and status reads, which the bundled proxy sends to the Mosart Web Apps host so that it can check the device's pairing and add the key: `/mosart/api/*/command` and every path beneath it (a Control Command is one level deeper, `/mosart/api/*/command/controlcommand/<name>`), `/mosart/api/*/timeline`, `/mosart/api/*/status` and `/mosart/api/*/assets/graphics`, with the `/mosart` prefix removed. A proxy of your own must do the same, or paired devices' commands are refused. On an installation with several studios, do the same for each studio's `/mosart/<studio>/api/*/command` and every path beneath it, `/mosart/<studio>/api/*/timeline`, `/mosart/<studio>/api/*/status` and `/mosart/<studio>/api/*/assets/graphics` paths: remove the `/mosart/<studio>` prefix and add the header `X-Panel-Studio` with that studio's ID. Without the header the command goes to the default studio, and devices paired with another studio are refused.
4. **Is authentication misconfigured?** If OIDC is on, an expired or rejected JWT causes the page to reload. Persistent reload loops mean the identity provider is rejecting the token. Re-check the OIDC settings.
5. **Check the browser console.** Open the browser developer tools (F12), reload Smart Panel, and look at the **Console** tab for red error messages. They usually point directly at one of the causes above (server unreachable, authentication rejected, network blocked).

A sign-in dialog appears over a panel

Where authentication is enabled, a panel normally holds its session for as long as its page is open. A **You will be logged out soon** countdown, or a **Session timeout!** dialog, means either that Smart Panel's switch has been turned on in **Allow session timeout**, or that the browser could not reach the Mosart Web Apps host for longer than the session length. Check that switch first, in **Server Configuration**, then the connection to the host.

The Live element reports a problem

The source picker states which part failed, so read the message before changing anything:

Message	What to do
NDI isn't available on the Mosart Web Apps server	Install the NDI Runtime on the Mosart Web Apps host.
Can't reach the NDI source service	Check that Mosart Web Apps is running.

Copyright © 2026 Vizrt

Message	What to do
No NDI sources found	The runtime is fine and nothing is advertising. Check the sources are powered on and on the same network as the Mosart Web Apps host.
TriCaster is not configured	Set the TriCaster host, username and password in the Configuration Tool.
TriCaster rejected the credentials	Correct the username and password in the Configuration Tool.
Can't reach TriCaster	Check the TriCaster is powered on and reachable on the network.
No TriCaster sources found	The TriCaster answered but advertised no inputs or outputs. Check the TriCaster's own configuration.
Can't load this video	For an HTML video source, check the link, that the file exists, and that the format is one the browser supports.
Enter a ws:// or wss:// OvenMediaEngine signalling URL, or an http:// or https:// stream/video URL	The address entered under **Stream** source does not start with a scheme Smart Panel accepts. Correct the scheme.

Buttons do not execute

If timers and labels work (no Offline chip) but a Button click does nothing, check:

- The button has an interaction (**Add interaction**), its **Command** is filled in and, for Template buttons, a template is selected.
- For a TriCaster, Media Sequencer or Viz Arc button, that the matching integration is configured. See *Integrations for Button Actions* above.
- **Authentication mismatch between Mosart Server and Mosart Web Apps.** Smart Panel needs the same authentication mode on both sides:
    - **If Mosart Server is configured with OIDC**, Mosart Web Apps must also have HTTPS and OIDC enabled (**Server Configuration** in the Configuration Tool).
    - **If you want to run over HTTP** (not recommended), Mosart Server must have a REST API key configured (RestApiKey), the same key must be entered as the **Authorization Key** in **Server Configuration**, and each device must be paired. A panel whose components are locked says *Your panel components are locked*; see *Pairing Devices* above. An older panel link that carries the key as a URL parameter (?apikey=<key>) still works while the key is right, but it exposes the key in browser history and logs.
    - **Locked although the device is paired?** The strip says why. *This device is paired with another studio.*: open a panel of the studio the device was paired for, or pair it with a code for this panel's studio.

Copyright © 2026 Vizrt

Mosart refuses the API key set in the Config Tool.: the **Authorization Key** in **Server Configuration** does not match the Viz Mosart server's RestApiKey; pairing again does not help, so correct the key.

- To confirm the underlying REST endpoint is responding, open the Mosart Server Swagger page in a browser:
  - *HTTP:* `http://<mosart-server-host>:55167/swagger/index.html`
  - *HTTPS:* `https://<mosart-server-host>:55168/swagger/index.html`
    If the Mosart Server is configured with a REST API key (RestApiKey), click Authorize in Swagger and supply the same key (header X-API-Key) before invoking any endpoint. If neither an API key nor OIDC is configured on the Mosart Server, the REST API is not usable, the Swagger page may load, but every call returns 401.

Layout not saved between sessions

Layouts are stored in a single file, *smartPanelLayouts.json*, under *C:\ProgramData\Mosart Medialab\Mosart Web Apps*. The built-in default layouts are part of the Mosart Web Apps installation and cannot be lost. If user-created layouts disappear after a server restart, check that this folder is included in your normal backup routine and has not been wiped by a reinstall.

Copyright © 2026 Vizrt

11 Template Editor (Early Access)

The new **Template Editor** offers an improved experience for creating, editing and managing templates.

Tip: Watch a 2' introduction here.

This guide is designed for both first-time users of Viz Mosart and experienced users of the old Template Editor, providing detailed explanations of the features, workflows, and recommended best practices.

11.1 Info: Early Access

The new Template Editor is currently provided as an **early access** version, intended for customers to test, explore, and provide insights and feedback for future versions.

11.2 Safety first

This version is **not supported for production use**. We recommend enabling it in a staging or test environment to safely explore its features and to always back up your channeltemplates.xml file *before* you start.

11.3 Template storage

Once enabled, the new Template Editor shares the same template storage configuration as the NRCS Plugin.
Only **file-based** template storage is supported in this version. Support for template databases will follow in a later version.

11.4 Control access

As a web-based app, the new Template Editor is available to *all* users on the local network. To control access, this version also offers **authentication**.
Once configured, the permissions apply to *all* Viz Mosart Web Apps.

11.5 Template Editor

The Template Editor is your tool for creating, modifying, and managing templates. Each template defines a carefully timed sequence of automated commands, such as camera switching, audio adjustments, graphic transitions, lighting control and much more. By designing robust and powerful templates, you can produce complex live broadcasts reliably and consistently, planned in a rundown or recalled manually at any time.

11.6 Benefits of the new Template Editor

- **Web-based**: Accessible from a browser on any computer on the network.

Copyright © 2026 Vizrt

- **A dashboard for the whole library**: Template sets are browsed as a list, a tree or side-by-side columns, with each set's templates and their inheritance laid out in a table, so the structure of the library is visible rather than implied.
- **Inheritance made visible**: A set's inheritance chain is shown level by level, and each template is annotated as inherited, a local addition, or an override of an inherited version, so you can see at a glance where a template's behaviour actually comes from.
- **Timeline view**: Gives a clear visual overview of all commands and their precise timing.
- **Sequences**: A sequence editor for building named, timed series of items on their own timeline.
- **Simplified command model:** Fewer, more powerful commands, to make template creation easier.
- **Easier to learn**: A clean, intuitive UI and keyboard shortcuts throughout lower the learning curve, even for new users.

11.7 How the new Template Editor Works

The new editor is designed to be compatible with existing Mosart templates, while introducing a more modern and user-friendly editing experience. Here's what you need to know:

- Template Editor directly modifies the same `channeltemplates.xml` file as the old editor.
- The data format is the same, so changes are backwards compatible. Both editors can be used side by side.
  - <custom data-type="emoji" data-id="id-0">:warning:</custom> However, the two editors are only partly synchronized:
    - Saving in the *old editor*: The template open in the *new editor* reloads automatically, unless it has unsaved changes. To see other changes, such as a new or renamed template, refresh the browser.
    - Saving in the *new editor*: Close the *old editor* (the server ignores an update while it is open), then press **Update server** in the template title bar. The Viz Mosart server reads the saved templates again, so the changes take effect in a rundown and show in the *old editor* without restarting AV Automation. This needs Viz Mosart 5.18 or later; with an older server, restart AV Automation instead.
  - There are no conflict or collaboration features: The latest save wins.
- This version supports a limited set of commands:
  - When opening an existing template, only the supported commands are visible.
  - When saving, unsupported commands are preserved, even though you do not see them. They are not lost.
  - All commands still appear in the old editor.
- Commands have been simplified, but may take some time to get used to:
  - The previous distinction between *Device Commands* and *Control Commands* is gone. Now you see a unified list.
  - Some commands have been unified for simplicity. For example, the command **Set source** can set a *Keyer*, *AUX* or *Crosspoint*, depending on which properties have been configured.
  - This change is only *visual*. When saving a **Set source**, the command is written to the file in one of the forms the old editor understands: a **SET_CROSSPOINT** control command, a *Keyfill*, or an *Auxiliary*, depending on the region, layer, channel and trigger. All three are visible in the old editor.
  - For more details, see Transitioning from the old Template Editor.

Copyright © 2026 Vizrt

11.8 Further Information

Additional documentation and guidance will continue to be added and refined as the new Template Editor evolves. For the latest updates, consult the Vizrt Documentation Center. If you require assistance with setting up Template Editor, please reach out to your Viz Mosart Support contact.

Copyright © 2026 Vizrt

11.9 Working with Template Editor

The Template Editor has two working surfaces: the **dashboard**, where you browse galleries, template sets, templates and sequences and see how sets inherit from one another, and the **editor**, where you build one template's commands on a timeline.

- The Dashboard
  - Browsing template sets
  - The set summary
  - The templates table
  - Sequences
  - Managing sets
- The Editor
  - Template title bar
  - Commands list and timeline
  - Audio fader panel
  - Command properties
- Keyboard Shortcuts
  - Templates table
  - Template sets
  - Template editor
  - General

11.9.1 The Dashboard

The Template Editor opens on the dashboard, headed **Galleries**. On the left are the template sets, and on the right the contents of the selected set.

Browsing template sets

The sets pane can be laid out three ways, chosen from its view options:

- **List view**: a flat list of sets, filtered as you type with the **Search sets** field, which sits beside the view options and filters all three views.
- **Tree view**: sets nested under the sets they are based on, so the inheritance structure itself is the navigation. Folders open only when you toggle them, and **Space** toggles the selected folder.
- **Column view**: each level of the inheritance chain as its own column, side by side, in the style of a column file browser.

The sets pane is resizable, and each set row carries its own actions for editing or deleting the set.

The set summary

Selecting a set shows a summary card before you open anything:

- **Inheritance chain**: the set's ancestry, level by level, from the set itself back to the root, with the current set marked.

Copyright © 2026 Vizrt

- **Template breakdown**: how the set's visible templates divide into *Inherited (unchanged)*, *Overrides (shadow parent)* and *New local (no parent version)*, with the total visible.

This answers the everyday questions about a set, namely what it adds, what it changes and what it merely passes through, without opening a single template.

The templates table

The selected set's templates are listed with their title and description, main identifier, status, created and last-edited dates. A search field filters the table, and each row's annotation tells you where the template comes from: inherited, an override, or local to this set.

- **Creating a template**: **Create new** asks for a title, a type and an optional description. The combination of type and title must be unique within the set.
- **Editing an override safely**: a template that overrides an inherited one behaves as you would hope. Renaming it, or changing its type, saves a copy in the current set and keeps the inherited template. Deleting the override restores the inherited version from the parent set, and the confirmation dialog says exactly that, naming the set the inherited version comes from.
- **Hiding a template**: the Template Editor has no per-template **Hide from user** switch. The **Hide from user** switch in the set properties dialog applies to the whole template set, and a set can block an inherited template entirely. A row marked *Hidden in this set* means playout skips it there, and the inherited version is blocked in that set.

Sequences

Next to **Templates**, the dashboard has a **Sequences** tab. A sequence is a named, timed series of items, each with an in-point and a duration in frames, built on its own timeline.

- **New sequence** creates one in the selected set; a search field filters the list.
- Open a sequence and add items with **Add item...**. Each row carries its **In (ff)** and **Dur (ff)**, and items are reordered by dragging, or with **Move up** and **Move down**.
- On the timeline, drag an item to adjust its offset and drag its edge to set its duration, zooming in and out as needed. A **Graphics out** marker sets where graphics are taken out.

Managing sets

Add set creates a new template set: a name, and optionally the set it is **Based on**, which is what places it in the inheritance tree. Recall settings for the set are grouped in the same dialog, and a template whose recall is not wired up is flagged with a warning that it will not recall on air unless a matching direct take exists. Deleting a set removes the templates in it, and the confirmation says so; the deletion cannot be undone.

11.9.2 The Editor

Opening a template takes you to the editor. **H**, or the back control, returns to the dashboard.

Template list: the editor keeps a template list on the left, headed **Templates**, with the back arrow to the dashboard and a **Create new** button. The button beside **Create new** undocks the list; while it is undocked, hovering the menu icon at the top left slides it back in. Selecting another template in the list opens it in the editor, and if the current template has unsaved changes you are asked whether to save them first.

Copyright © 2026 Vizrt

Template title bar

The header shows the selected template's name, variant, description and location, and carries:

- **Mix Delay**: a frame-based delay applied to specific command types (main source switches, region source switches, audio fades) when the template is taken to air, for fine-tuning on-air action timings. While Mix Delay is set, the individual offsets of the commands it applies to are ignored. A Set Source on a numbered layer is not one of them and keeps its own offset, unless it was created in the desktop Template Editor. Mix Delay becomes available once the template has a Switch Main or Switch Region command.
- **Save**, and a menu with **Edit** (rename, change type, update the description or the Mix Delay value) and **Delete**.
- **Update server**: makes your saved changes take effect in a rundown without restarting AV Automation. The Viz Mosart server reads all saved templates again, not only the open one. Once a show has started, the server holds the update until the rundown is next reloaded, updated or recued with the timeline stopped, or until the server goes idle. The button is disabled while a save is being written, and while the template has unsaved changes; its tooltip then reads *Save your changes first, then update the Mosart server.*
  Close the desktop Template Editor on the server first: the server ignores the update while it is open. The button needs Viz Mosart 5.18 or later; if the server cannot be reached, or is older, an error says its templates are unchanged. The confirmation *Update sent to the Mosart server.* means the request was delivered, not that the server has finished reading the templates.
- The **?** help menu: **Keyboard Shortcuts**, **Documentation**, **What's new**, **Give feedback** and **About**.

Commands list and timeline

The **Commands** panel presents two complementary views, switched between **Timeline** and **List**. Both are interactive and synced with the **Command properties** panel.

- **Timeline view**: a visual representation of when each command executes during the *Cue*, *Take* and *Out* phases, showing offsets and transition lengths. Multi-phase commands, such as *Switch Main* or *Switch Region*, are visible across the phases they act in. Offsets and transition rates can be adjusted by dragging directly on the timeline.
- **List view**: the same commands and their core properties in a compact linear layout, for quick review when precise timing visualization is not required.

> **Info**: Commands are grouped by device family, not by execution sequence. This reflects the structural grouping within a template, but not necessarily the order Viz Mosart executes them at runtime.

Adding commands

1. Click the **+** button at the top right of the **Commands** view to open the command library.
2. Browse by device family:
   - **Switcher**: Switch Main, Switch Region, Set Source, Macro Recall and EMEM Recall
   - **Audio Mixer**: Set Fader and Control Audio Input
   - **Video Wall**: Video Wall
   - **Robotic Camera**: Camera Shot
   - **Light**: Light Control

Copyright © 2026 Vizrt

- **GPI/O**: GPI/O Control
- **Weather**: Weather
- **Graphics**: Graphics
- **Generic REST**: Generic REST

Each command entry has a short description, with **View More** for details on the longer ones.

3. Select a command and click **Add**. Editing continues immediately in **Command properties**.

Tip: To add several commands in a row, hold **Shift** when you click **Add**. The dialog stays open so you can keep going.

Info: Some commands can be added only a limited number of times to one template, and once a command has reached its limit its **Add** button is disabled in the library. Switch Main, Switch Region, EMEM Recall, Video Wall, Weather and Graphics can each be used once per template; Macro Recall can be used twice, one per bus, and Light Control twice, one per trigger. The other commands are unlimited.

Info: A command created in the desktop Template Editor may carry an annotation that some behaviours, such as how Mix Delay applies, can differ. See Transitioning from the old Template Editor.

Audio fader panel

The **Audio Fader** panel gives a quick overview of all **Set Fader** commands in the template, showing each fader's number, its trigger, its audio input (for example *MIC 1*) and its level in dB. Adjust a level by dragging the fader directly, or double-click the fader knob to set it to *0* dB. Select the command in the list or timeline to change its other parameters, such as the fade up trigger. **Control Audio Input** commands set no level and do not appear in the panel.

The **L1** and **L2** buttons at the left of the panel choose which level the faders edit: **Level 1**, the default, or **Level 2**. A small marker beside each fader shows the other level, and you can drag the marker to change that level without switching. The level shown in the command list and on the timeline follows the same choice.

The bolt button on a card turns green when switcher commands are linked to that fader, and hovering over it lists them. To link commands to a fader, see **Set Fader** in the Command Reference.

Command properties

The **Command properties** panel shows the configurable options for the selected command. Which properties appear depends on the command and the device it targets; the full set is documented in the Command Reference.

11.9.3 Keyboard Shortcuts

Press **K** anywhere, or pick **Keyboard Shortcuts** from the **?** menu.

Copyright © 2026 Vizrt

Templates table

Action	Shortcut
Move selection up / down	`↑` / `↓`
Jump to first / last template	`Home` / `End`
Open selected template	`Enter`
Open in new tab	`Ctrl` + `Enter`
Edit selected template	`F2`
Delete selected template	`Delete`

Template sets

Action	Shortcut
Move selection, collapse / expand	`↑` / `↓`, `←` / `→`
Edit selected set	`F2`
Delete selected set	`Delete`

Template editor

Action	Shortcut
Create new template	`Ctrl` + `Alt` + `N`
Delete template	`Ctrl` + `Delete`
Add command	`Ctrl` + `Alt` + `C`
Edit template properties	`Ctrl` + `Shift` + `E`

Copyright © 2026 Vizrt

Action	Shortcut
Edit template mix delay	`Ctrl` + `Shift` + `↑` / `↓`
Save template	`Ctrl` + `Shift` + `S`
Back to dashboard	`H`

General

Action	Shortcut
Show keyboard shortcuts	`K`

Info: In this version, only file-based templates are supported. The galleries offered in the picker at the top of the dashboard are the ones defined in that channeltemplates.xml file. Support for gallery-based template databases will be introduced in a future release.

Copyright © 2026 Vizrt

11.10 Setup and Administration of TE

The new Template Editor is part of the **Mosart Web Apps** package. It is accessed and managed through the Configuration Tool. **LAUNCH CONFIG TOOL** on the installer's last screen opens it, and the desktop and Start menu shortcuts open it later.

1. **Open the Configuration Tool.**
   On an installation that has not been configured yet, the Configuration Tool opens the Web Apps quick setup wizard. Finish it, or choose **Skip setup**, to reach **Home**, which shows all available applications. Template Editor is listed as one of the applications, but must be enabled before it becomes accessible.
2. **Enable Template Editor.**
   a. Go to **App Configuration**.
   b. Select **Template Editor**.
   c. Review the important notes shown there before enabling. The application is in early access, and the notes describe its current limitations.
3. **Save and launch.** Once you have enabled the Template Editor:
   a. Click **Save**. The change applies at once, with no restart.
   b. Once you have saved, the **Open** button becomes available. Click **Open** to launch Template Editor in your default browser, at `/templateeditor` on the Mosart Web Apps host.
4. **Confirm the template storage file location.**
   Template Editor shares the template storage path configured on the **NRCS Plugin** page.
   - Ensure the **Local** file storage path is valid and accessible.

> **Info**: Templates stored in a *database* are not supported in this version. Only file-based storage works with the Template Editor.

5. **Control access.**
   As a web-based app, the new Template Editor is available to all users on the local network.
   - To control access, this version also offers **authentication**. Once configured, the permissions affect *all* Viz Mosart Web Apps.

> **Note**: The Template Editor writes to the same `channeltemplates.xml` that your production uses. Back the file up before exploring, and prefer a staging environment. See the safety notes on Template Editor (Early Access).

Copyright © 2026 Vizrt

11.11 Command Reference

This guide covers command function, parameter structure, and how they are used in typical workflows.

- Video Switcher Commands
  - Command Structure
    - Region
    - Layer
    - Channel
    - Source
  - Switch Main
    - Properties
  - Switch Region
    - Command properties
  - Set Source
    - Command properties
    - Examples
  - Macro Recall
    - Command properties
    - On-air protection
- Audio Mixer Commands
  - Set Fader
    - Command properties
  - Control Audio Input
    - Command properties
- Robotic Camera Commands
  - Camera Shot
    - Command properties
- Light Commands
  - Light Control
    - Command properties
- GPI/O Commands
  - GPI/O Control
    - Command properties
- Weather Commands
  - Weather
    - Command properties
- Graphics Commands
  - Graphics
    - Command properties
- Generic REST Commands
  - Generic REST
    - Command properties
- Fine-tuning On-Air Action Timings
  - Delay mechanism summary

Copyright © 2026 Vizrt

- Template Terms and Definitions
- Commands in this description are organized by the *type* of device they control (for example a *switcher* or *audio mixer*) with examples illustrating their common usage scenarios.
- For users familiar with the existing *desktop Template Editor*, a separate section, Transitioning from the old Template Editor, highlights key differences in behavior and compatibility.
- As the Template Editor evolves, more command types will be added.

11.11.1 Video Switcher Commands

Command Structure

For video switcher (vision mixer) commands that assign a source, such as *Switch Main*, *Switch Region*, or *Set Source*, Template Editor uses a hierarchical structure that precisely defines source destination, in the form:

Region → Layer → Channel → Source.

This hierarchical model supports a wide range of production systems, from traditional hardware switchers to modern software-based mixers. While it is essential for source-mapping commands, not all vision mixer commands necessarily use this structure.

Here's a breakdown of each element:

Region

The top-level destination for the source. This could be:

- **Program/Preview** *(PP)*: The main switcher output
  - **Mix/Effect** *(ME1, ME2, etc.)*: For complex layer compositions
  - **Auxiliary** *(AUX)*: For independent outputs
    In some systems, this is called either a *scene* (Panasonic Kairos), or a *graphic scene* (see Viz Engine as a Switcher).

Layer

Identifies a compositing layer (for example, *Background*, *Layer 1*, *Layer 2*, *Layer 3*) where the source should be placed.

- The *Background* (also referred to as *Main*) layer represents the base layer of the visual composition, the foundation onto which all other visual elements (graphics, overlays, picture-in-picture, etc.) are added.
  This layer may support A/B transitions (Preview + Program). In traditional switchers, this corresponds to the background video source of any M/E bank, not just the Program output.
  Each M/E has a base input (the *Background*), which can have keys and effects stacked above it. In modern, layer-based switchers like Panasonic Kairos, it serves as a virtual construct to simulate this foundation. The web-based Template Editor uses this concept to unify how the base layer is addressed across both traditional and fully composited switcher architectures.
  Note: This layer is *not* available when region *Auxiliary* is selected.
- *Layer 1, Layer 2, Layer 3*, and so on, are used to place content on top of the main background layer. Each of these layers currently has just one video bus (called *A*) and currently does not support transitions between

Copyright © 2026 Vizrt

two sources like traditional A/B transitions.
Examples are:

- Panasonic Kairos supports multiple layers in both *single* and *dual-bus* modes.
- TriCaster M/Es allow multi-row compositions (A, B, C, D).
- Viz Engine treats superchannels as layers with A/B buses.

The *Layer* may be fixed based on command type (for example, *Switch Main* does not expose a layer setting, this is set by default to *Background* and cannot be changed).

Channel

This is the final destination where the source is assigned. In a dual-bus background layer, this might be A (program) or B (preview). Single-bus layers typically only use channel A.

Source

The actual input, such as a camera, live feed, graphics input etc., referenced by its name (crosspoint) configured on the Viz Mosart server in AV Automation in the menu
Devices > A/V Setup > Video Config. (for example, *CAM1*, *LIVE1*).

Switch Main

The **Switch Main** command sets the source on the main output (*Program*/*Preview*) of the switcher. This can be a direct source like a camera, graphic, or video feed (for example, *CAM1*), or a secondary region (for example, *ME1*) that is being composed elsewhere.

Properties

- **Region**: Program/Preview (the main output). This can not be changed.
- **Source**: The desired source (e.g., CAM1). *Default*: the first source in the list.
- **Trigger**: Specifies where the command is executed. This can not be changed. The command transitions the selected source on the main output from Preview (B) to Program (A) when the template is taken on air. In the Properties on the command on TAKE is visualized, but in the Timeline view, both the commands on CUE and on TAKE are visualized.
- **Offset**: Optional delay in frames. Ignored when Mix Delay is enabled; instead, the Mix Delay applies a unified offset to this and other commands. *Default*: 0 frames.
- **Transition style**: CUT, MIX, WIPE or EFFECT. CUT is in fact a MIX with transition rate 0 frames. *Default*: CUT
- **Transition rate**: MIX and WIPE support optional transition rate (in frames). *Default*: 0 frames.
- **LINK COMMAND**: choose the **Set Fader** command this command is linked to, or *None*. See **Set Fader**.

Switch Region

Switches a source to the background of another mixer region (for example, *ME1*, *ME2*) different from the main output (PP). This allows for switching sources in parallel with the main output - typically used in compositions, picture-in-picture, or secondary outputs.

Command properties

- **Region**: ME1, ME2, etc.

Copyright © 2026 Vizrt

- **Layer**: *Background*. Like **Switch Main**, **Switch Region** does not expose a layer setting. It always switches the background layer of the selected region.
- **Source**: The desired source (for example, *CAM1*). Default: The *first* source in the list.
- **Trigger**: Specifies where the command is executed. This cannot be changed. The command transitions the selected source on the mixer region from Preview (B) to Program (A) when the template is taken on air.
  In the command Properties view only the TAKE command is shown, but in the Timeline view, both the commands on *CUE* and on *TAKE* are visualized.
- **Offset**: Optional delay in frames. Ignored when Mix Delay is enabled; instead, the Mix Delay applies a unified offset to this and other commands. Default: *0* frames.
- **Transition style**: CUT, MIX, WIPE or EFFECT. CUT is in fact a MIX with transition rate 0 frames. Default: *CUT*.
- **Transition rate**: MIX and WIPE support optional transition rate (in frames). Default: *0* frames.
- **LINK COMMAND**: choose the **Set Fader** command this command is linked to, or *None*. See **Set Fader**.

Info: Limitations with current version

- A maximum of *two* switch commands are supported in a single template execution
  - one for PP (main) and
  - one for a secondary region (for example, ME1).
- Only *one* non-main region (ME1, ME2, etc.) can be used for **Switch Region**.
- Shared *transition* and *offset*: Both **Switch Main** and **Switch Region** commands in a template will use the *same* transition type and *same* offset.

Set Source

The **Set Source** command is used to assign a specific source to a *single-bus layer*, typically used for *keyers*, *picture-in-picture*, *video walls*, ***auxiliaries*** or other compositional elements within a production scene.

Unlike **Switch Main** or **Switch Region**, which operate on background layers and typically trigger a transition (for example, *Mix*, *Cut*), **Set Source** is a direct source assignment. One case is an exception: when the **Region** is not *Auxiliary*, the **Layer** is not *Background*, the **Channel** is *A* or empty and the **Trigger** is *TAKE*, the command also carries **Transition style** and **Transition rate**, with the same choices as **Switch Main** and *CUT* as the default. Commands created in the desktop Template Editor never show them.

Command properties

- **Region**: Specifies the mixer region where the source should be applied. For example, *Program/Preview* (PP), *ME1*, *ME2*, *Auxiliary*, etc.
- **Layer**: Indicates the layer within the selected region. Can be *Background* (except when **Region** is set to *Auxiliary*) or a numbered layer (for example, *Layer 1*, *Layer 2*, etc.).
- **Channel**: Defines the channel to assign the source to, A or B. This is only applicable when the **Layer** is set to *Background*.
- **Source**: The source to be assigned (for example, *CAM1*, *LIVE1*). If left empty, the first available source in the drop-down list is used by default.
- **Trigger**: Specifies when the command should be executed, on *CUE*, *TAKE*, or *OUT*. Only a single *OUT* command per template is supported in this version, and an information line below the **Trigger** field says so. If you duplicate or paste a command whose trigger is *OUT* into a template that already has one, the copy is moved to *TAKE*.

Copyright © 2026 Vizrt

- **Offset**: An optional delay (in frames) before the command is executed. It can be set when the **Trigger** is *CUE*, and also when the **Trigger** is *TAKE* and the **Layer** is not *Background*, provided the command was not created in the desktop Template Editor. In every other case the field is disabled and a hint explains why.
- **LINK COMMAND**: choose the **Set Fader** command this command is linked to, or *None*. A command that cannot be linked shows a note there instead. See **Set Fader**.

Info: The **Mix Delay** setting (defined at the template level) applies to the **Set Source** command only if:
- the **Layer** is set to *Background*, and
- the **Trigger** is set to *TAKE*.

Examples

- **Assigning keyers for a 2Box composition**.
  This template uses two **Set Source** commands to assign video sources to different keyer layers on the *ME1* region.
  - The first command maps *CAM1* to *Layer 1*, used as the left video box.
  - The second command maps *LIVE2* to *Layer 2*, used as the right video box.
    This setup is commonly used in 2Box or split screen layouts, where two speakers or feeds need to be shown side by side, with the ME handling the composition.
- **Taking multiple sources simultaneously**
  This template prepares and takes multiple sources simultaneously: It sets LIVE1 to preview (channel B) of ME2 on cue, then switches LIVE1 to program (channel A) of ME2 and routes Aux1 to an auxiliary layer when the template is taken.

Macro Recall

The **Macro Recall** command recalls a switcher macro on one bus, chosen by the trigger. A CUE command recalls on the preview bus when the template is cued; a TAKE command recalls on the program bus when the template is taken.

Command properties

- **Recall preview** (shown when **Trigger** is CUE) or **Recall program** (shown when **Trigger** is TAKE): the macro to recall. With a TriCaster configured this is a picker driven by the TriCaster macro folder; otherwise it is a free-text field.
- **Function**: what the switcher does with the macro. **Recall**, **Take**, **Recall and Take**, or **Load project**.
- **Trigger**: CUE or TAKE. Only one Macro Recall command can own each bus, so once a template already holds the other one, this becomes read-only. Deleting that command frees the trigger again.
- **On-air protection**: see below.
- **Test action** recalls the macro immediately, so you can check it while building the template.

On-air protection

On-air protection sits in the command's **EXECUTION CONTROL** group, directly under **Trigger**. While the template holding the macro is on air, it stops macros belonging to other templates from firing when those templates are cued, so the on-air look is not disturbed.

Copyright © 2026 Vizrt

Value	What it blocks
Off	Nothing. This is the default.
All templates	Every other template.
Same template type	Only other *protected* templates of the same type, for example another DVE.

Taking a template to program always runs that template's own macro, so the next look still goes on air. Protection holds back the macros of templates that are merely being cued behind it.

Info: The value belongs to the template's macro recall as a whole, not to one bus. If a template holds both a CUE and a TAKE **Macro Recall** command, changing **On-air protection** on either one changes it on both.

11.11.2 Audio Mixer Commands

Set Fader

The **Set Fader** command is used to adjust the audio level of a specific source at a defined point in the template timeline.

Command properties

The properties are grouped in four sections in the panel.

FADER PROPERTIES

- **Audio input:** The audio input to be controlled (e.g., Mic1, Mic2, etc.) as configured on the Viz Mosart server in AV Automation in the menu
  Devices > A/V Setup > Audio Config. *Default*: the first audio input in the list. When commands are linked to the fader, a note under the field says so: on air, the fader follows the audio channel that A/V Setup maps to the linked source, not the audio input chosen here.
- **Level 1:** The target audio level in decibels (dB). The fader scale runs from *+10* dB at the top down to *-90* dB at the bottom, where the level reads as minus infinity. A typed value is rounded to a whole dB, and a value outside the scale is set to the nearest end of it. A newly added **Set Fader** command starts at *0* dB.
- **Level 2:** An alternative level, used when playout runs in level 2 mode, for example under a voice-over. Same scale as **Level 1**. To set it on the fader cards, choose **L2** in the Audio Fader panel. *Default*: *-90* dB.
- **Level 3:** A third level, on the same scale. *Default*: *-90* dB.
- **Loudness preset**: the preset number recalled on a Junger loudness processor. *0*, the default, means no preset.
- **Input device**: what feeds this channel: *None*, the default, *Video server*, *Video server Ch2*, *Microphone*, *Graphics*, *External source*, *Telephone* or *Soundplayer*. Playout uses it to find the fader for a video server clip, graphics or an external source.
- **Mode**: the channel mode set on the mixer: *None*, the default, *Stereo*, *Mono*, *Left* or *Right*. Used by Studer Vista and Lawo consoles.

Copyright © 2026 Vizrt

CROSSFADE SETTINGS

- **Fade in**: how many frames the fader takes to reach its level when the command fires. *Default*: 4 frames.
- **Fade out**: how many frames the fader takes to close when the template leaves air. *Default*: 4 frames.

EXECUTION CONTROL

- **Keep level**: *Enable* keeps the fader at its level when the next template goes on air, instead of closing it. *Default*: *Disable*.
- **Fade up trigger**: *CUE* fades the channel up while the template is in preview. *TAKE* fades it up when the template goes on air. *Default*: *TAKE*.
- **Audio signal delay**: a fixed signal delay, in milliseconds, set on this fader in the mixer hardware. It does not delay when the command runs. Used only by Studer Vista, Studer 3000 (Ember) and Lawo consoles. *Default*: 0 ms.
- **Manual take down**: *Enable* leaves the fader open when the template leaves air, for the operator to close. *Default*: *Disable*.

LINKED COMMANDS

This section lists the switcher commands in the template that can be linked to the fader, each with a switch. Turn a switch on to link that command to this fader. Click the section heading to collapse or expand the list.

- **Switch Main** and **Switch Region** share one link, so they appear as one row, and linking either one links both.
- Each **Set Source** command has a row of its own, named after its source. A **Set Source** command can be linked only when its **Layer** is not *Background*, its **Channel** is *A* or empty and its **Trigger** is *TAKE*, and only if its properties do not show the note that it was created in the desktop Template Editor. **Channel** is shown only while the **Layer** is *Background*. If a command that meets the other conditions still cannot be linked, set its **Layer** to *Background*, choose **Channel** *A*, then choose the layer again.
- A command is linked to one fader at a time. Turning it on here moves it from the fader it was linked to before, and the row shows which fader it came from.

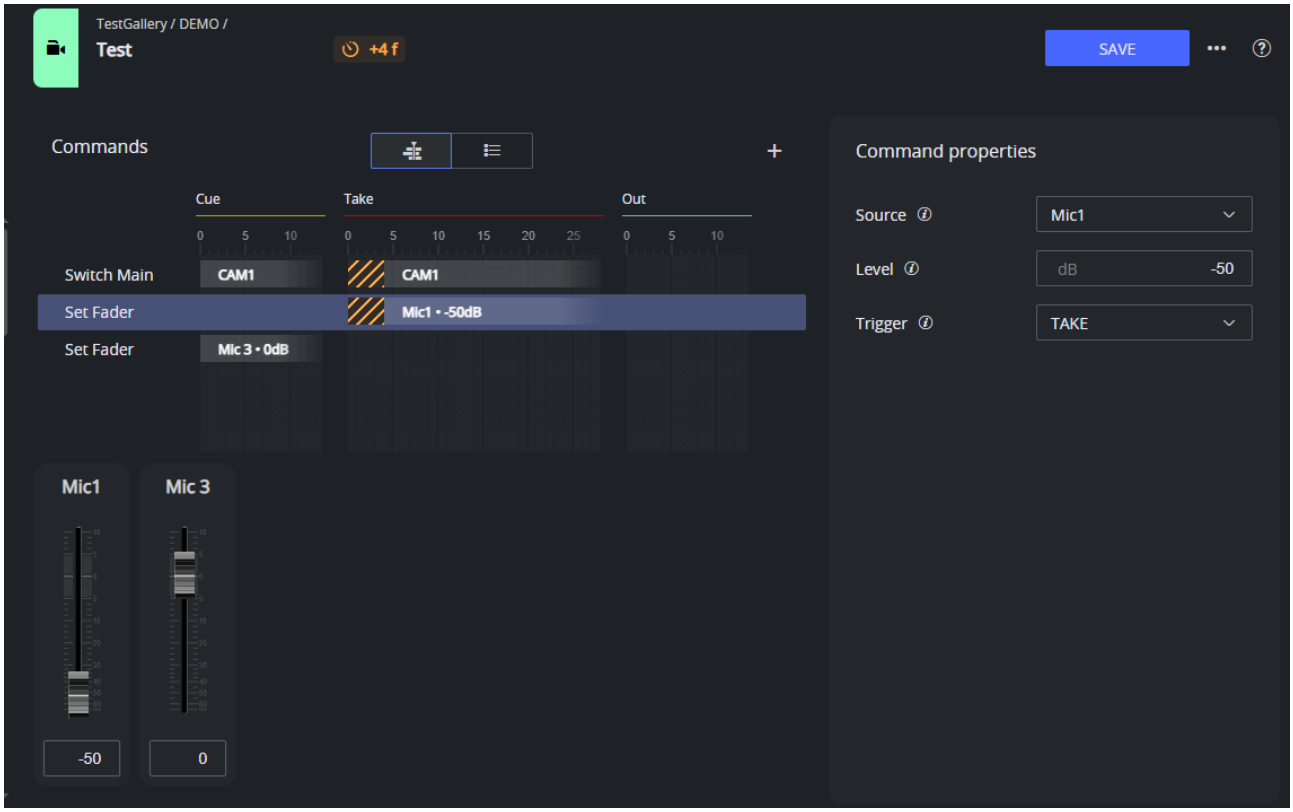
You can also link from the switcher command itself. Select **Switch Main**, **Switch Region** or **Set Source**, and in its **LINK COMMAND** section choose a fader, listed with its fader number and audio input, or *None* to unlink it. A **Set Source** command that cannot be linked shows a note there instead of the list.

A linked switcher command shows a green badge with the fader number in the command list and on the timeline. Hover over the badge to see which fader it is, and click it to select that **Set Fader** command and bring its card into view in the Audio Fader panel. On the fader card, the bolt button turns green, and hovering over it lists the linked commands.

Fader numbers follow the order of the cards in the Audio Fader panel. Deleting a **Set Fader** command unlinks the commands that were linked to it. A duplicated or pasted **Set Fader** command gets the first audio input that no other fader in the template uses, or no audio input when every input is already in use, and starts with no linked commands. A switcher command pasted into another template loses its link.

> **Info**: The **Mix Delay** setting (defined at the template level) applies to the **Set Fader** command only if its **Fade up trigger** is *TAKE*.

Copyright © 2026 Vizrt



Control Audio Input

The **Control Audio Input** command changes the state of one or more audio inputs when the template is taken. It puts the inputs in standby, so automatic audio leaves them alone, or releases them again, or it mutes or unmutes them. It sets no level: use **Set Fader** for that.

Command properties

The properties are grouped in three sections in the panel.

GENERAL

- **Control**: *State* or *Volume*, the group of actions that **Set to** offers. Choosing the other group selects its first action.
- **Set to**: what happens to the audio inputs when the template is taken. *Default*: *Standby*.
  - Under *State*: *Standby* puts the inputs in standby, so automatic audio leaves them alone. *Release standby* releases them again. *Release all* releases every audio input in standby, so it needs no input.
  - Under *Volume*: *Mute* mutes the inputs and *Unmute* unmutes them.

MANAGE INPUTS

- **Audio input**: an audio input the action applies to, chosen from the same list as the **Set Fader** audio input, as configured on the Viz Mosart server in AV Automation. A newly added command starts with the first audio input in the list.

Copyright © 2026 Vizrt

- **Add audio input** adds a row, set to the first audio input the command does not use yet. The button next to a row removes it. **Add audio input** is disabled when no audio inputs are configured.

With *Release all*, the section lists no inputs and shows a note that no input is needed.

EXECUTION CONTROL

- **Trigger**: always *TAKE*, and shown disabled. Viz Mosart runs audio input actions on take only.
- **Offset**: shown disabled. The action runs when the template is taken, with no offset.

In the command list the command shows its action and its audio inputs, and its icon shows the action. On the timeline its bar reads the same way, for example *Standby • MIC 1, MIC 2*. Both name the first three inputs and count the rest, for example *+2 more*.

Info: A template can hold any number of **Control Audio Input** commands. Each command has one action, so to put some inputs in standby and mute others, add one command per action.

Info: Fader actions set up in the desktop Template Editor (standby, release standby, release all, mute and unmute) open as **Control Audio Input** commands. Neighbouring audio channels with the same action open as one command.

Note: A command with no audio input does nothing, except with *Release all*. The editor flags it in the command list, on the timeline and in the properties panel (*No audio input. The command does nothing until an input is added.*). Nothing is saved for it, so it is gone the next time you open the template.

11.11.3 Robotic Camera Commands

Camera Shot

The **Camera Shot** command recalls a stored shot (a framing) on a robotic camera head. On *CUE*, the head pre-positions while the template is cued, normally before its camera is on air. On *TAKE*, it moves as the template is taken, and the move is visible on air.

Pages and presets are the numbers, or names, already stored in the camera controller. The Template Editor saves them as typed and does not check them against the controller.

Command properties

The properties are grouped in up to four sections in the panel.

CAMERA

- **Cameras**: *One camera* aims the shot at a single camera number. *All cameras* sends it to every camera that is not in standby.
- **Camera number**: which camera the shot is routed to, as configured on the Viz Mosart server in AV Automation under **Devices > Properties > Camera Robotics**. Shown for *One camera*.

SHOT

- **Shot name**: a free-text label for this shot. Playout never reads it, so it is there to make the template readable.

Copyright © 2026 Vizrt

- **Page**: what this addresses depends on the camera system. On NDI PTZ and similar it is the camera itself; on the others it is a preset page, show or matrix.
- **Preset**: the stored framing to recall, a number on some camera systems and a name on others. Free text, stored exactly as typed.
- **Move time**: how long the move takes. The unit depends on the camera system: seconds on Telemetrics, where **0** means cut straight to the framing, and a speed fraction between 0 and 1 on NDI PTZ.

EXECUTION CONTROL

- **Trigger**: *CUE* or *TAKE*, as described above.
- **Offset**: frames to wait after the take before the move starts. Only NDI PTZ and Vinten honor it (on Vinten, every shot waits for the **Offset** of the last shot that fires), and a shot that fires on CUE has no delay.
- **Protect on-air camera**: a video switcher crosspoint, for example `CAM1`. If that crosspoint is on air when the recall fires, the shot is skipped and the head stays where it is. Leave it empty for no protection.

WATCH CAMERA

- **Device** and **Video source**: an optional live picture to watch while you set the shot, from the TriCaster inputs (listed as *Switcher*) or from a machine sending NDI directly. **Video source** appears when the device has more than one. Sources whose stream cannot be received are not listed, and the section is hidden when none is found.

> ⚠ **Note**: A shot with no **Preset** is saved, and the editor flags it: *No preset set*. Set a preset before the template goes on air: a **Preset** you clear here is saved empty, and Telemetrics then recalls preset 0.

> ⚠ **Note**: Use one shot per camera per trigger. Two shots aimed at the same camera on the same trigger fire in list order without waiting for each other, so the last recall wins. The editor warns when a template does this.

11.11.4 Light Commands

Light Control

The **Light Control** command sends a command and a value to the light desk, for example `Go` with the value `22` to run cue 22. The editor does not check either field. Which commands work depends on the light controller set up in AV Automation: MIDI Show Control takes commands such as `Go`, and ETC Eos OSC and grandMA3 OSC accept only `Custom`, with an OSC message as the value.

Command properties

- **Command**: the command to send, for example *Go* or *Custom*, spelled as the light controller expects.
- **Value**: the value sent with the command, for example a cue number.
- **Trigger**: *CUE* sends the command while the template is cued, *TAKE* when it is taken.

Copyright © 2026 Vizrt

Info: A template holds **one light command per trigger**, so at most two: one on CUE and one on TAKE. A second light command goes to the free trigger, and **Trigger** then becomes read-only. Deleting one of them frees its trigger again.

Info: There is no **Offset** on a light command. The light desk gets the command at the trigger itself, and the field is shown disabled with a hint saying so.

Note: Fill in both **Command** and **Value**, otherwise nothing is sent. The editor flags a command with no **Value** (*No value set*), and that command is gone the next time you open the template.

11.11.5 GPI/O Commands

GPI/O Control

The **GPI/O Control** command lets a template switch something outside the automation chain, for example an on-air lamp. It fires an output port on the WebIO boxes: it pulses the port, or sets it high or low and leaves it there.

Command properties

- **Port**: the output port, *0* to *47*. Ports 0 to 11 are on the first WebIO box, 12 to 23 on the second, and so on across four boxes of twelve. A typed value outside that range is accepted but flagged.
- **Action**: what the port does when the command fires. *Pulse* closes the contact and releases it again. *Set high* closes it and leaves it closed, *Set low* opens it. Use high and low as a pair, for example a lamp switched on by one story and off by another.
- **Trigger**: *CUE* fires the port while the template is cued, *TAKE* when it is taken.

Info: A template can hold any number of GPI/O commands, and all of them fire on their trigger, in list order. Add one command per port: if two commands share a port and trigger, both are sent and the last one in the list decides the port's final state. Two pulses become two requests back to back, and *Set high* followed by *Set low* leaves the port low.

Info: There is no **Offset** on a GPI/O command. The port fires at the trigger itself, and the field is shown disabled with a hint saying so.

Note: A port outside 0 to 47 fires nothing, and the editor flags it in the command list, on the timeline and in the properties panel (*Port is outside 0 to 47*). A port above 47 is saved, and AV Automation logs an addressing error at the trigger. A negative port is not kept: it is gone the next time you open the template.

Copyright © 2026 Vizrt

11.11.6 Weather Commands

Weather

The **Weather** command plays a Viz Weather data item with the template. The data item is loaded when the template is cued and played when it is taken, or later on a trigger start.

Command properties

- **Concept**: the Viz Weather concept the data item is played in. A new command starts with *1*.
- **Data name**: the Viz Weather data item to play. A new command starts with *itemname*, so replace it with the name of your data item.
- **Wait for trigger start**: *Off*, the default, plays the weather when the template is taken. *On* means taking the template does not play the weather: it plays on a trigger start instead.

Both text fields are sent to Viz Weather exactly as you type them. The editor does not check what you type, it only flags a missing data name. The command list shows the concept and the data name, and the timeline shows the data name.

Info: A template holds **one Weather command**. Once it is added, its **Add** button in the command library is disabled.

Info: There is no **Trigger** and no **Offset** on a Weather command. The data item is always loaded when the template is cued. It plays when the template is taken, or on a trigger start when **Wait for trigger start** is *On*.

Note: Fill in **Data name**. A command with no data name is saved, but every take plays an empty Viz Weather item. The editor flags it in the command list, on the timeline and in the properties panel (*No data name set*).

11.11.7 Graphics Commands

Graphics

The **Graphics** command plays a fullscreen graphic on a graphics engine with the template. The graphic is loaded when the template is cued and taken when the template is taken. You set the engine and a default scene, and at playout the story item's graphic replaces that scene, unless the content source is *newsroom*.

Command properties

The properties are grouped in three sections in the panel.

GENERAL

- **Engine**: the graphics engine, usually *1* to *5*. Type *1/2* or *3/4* to ripple between two engines. A new command starts with *1*.

Copyright © 2026 Vizrt

- **Content source**: *dataconnection*, the default, *newsroom* or *none*. With *newsroom*, the scene comes from newsroom fields. With the others, the story item's graphic replaces the scene at playout.
- **Scene name**: the default scene, played when the story item brings no graphic of its own. Empty in a new command.
- **Scene text**: the default text for the scene. Empty in a new command.

EXECUTION CONTROL

- **Scene name is a database index**: *On* means the scene name is a database index rather than a scene path. Used by Vizrt, Xpression and Orad.
- **Continue on DVE forward**: *On* means a DVE FORWARD command sends a continue to this graphic while the template is on air.
- **Take out on leave**: *On* means the graphic is taken out when the template leaves air.
- **Take out delay**: how many frames to wait after the template leaves air before the graphic is taken out, from *0*, the default. Shown only while **Take out on leave** is *On*.

The three switches are *Off* in a new command.

BRAND OPTIONS

These options apply only to some graphics systems, and each one names the system it belongs to. Leave them as they are if your system is not named.

- **Use Viz Content Pilot**: *On* means the graphic is played through Viz Content Pilot. Vizrt only. *Off* by default.
- **Translate cue (Vizrt)**: how a cue for a Transition Logic element is translated: *Off*, the default, *No translate*, *Take* or *Take out*. *Off* leaves the setting out of the template.
- **Mixer effect (Adaptive Live Producer)**: the mixer effect the engine switcher keys on, for example `ME1`. **Engine** then holds the superchannel. Leave it empty for no mixer effect.

The command list shows the engine and the scene name, and the timeline shows the scene name.

Info: A template holds **one Graphics command**. Once it is added, its **Add** button in the command library is disabled.

Info: There is no **Trigger** and no **Offset** on a Graphics command. The graphic is always loaded when the template is cued and taken when it is taken, and the panel says so in **EXECUTION CONTROL**.

Note: With **Content source** set to *newsroom*, fill in **Scene name**. The story item's graphic does not replace the scene then, so a command with no scene name plays no graphic unless a newsroom field sets the scene name. The editor flags it in the command list, on the timeline and in the properties panel (*No scene name set*).

11.11.8 Generic REST Commands

Generic REST

The **Generic REST** command sends an HTTP request to a REST server when the template is cued, taken or taken out. You choose the server, the request method and the path, and add parameters, headers and a body as the call needs. Add one command per call.

Copyright © 2026 Vizrt

Command properties

- **Command title**: your name for the call. A new command starts with *Command*.
- **Assign to device**: the REST server the call goes to, picked from the servers the Viz Mosart server lists. Each server is listed by its title, or by its Device ID when it has no title. A disabled server shows *(disabled)* after its name. A server the list no longer holds stays on the command and shows *(not configured)*. When no servers are listed, because none are set up or the list cannot be read, the field becomes a text field, and you type the Device ID of the server, for example `genericrest-ix-0`.
- **Execute command on**: *Preview* sends the call when the template is cued, *Program* when it is taken, and *Out* when it is taken out. A new command starts with *Preview*. This is the command's trigger, so changing it moves the command on the timeline.
- **Relative Path**: the path added to the server's base URL, for example `path/to/resource`. It must not start with a scheme such as `http://` or with a slash, and it can carry a query string, for example `path/to/resource?param1=value1`. The info icon next to the label explains this. The field shows an error when the path is not valid or the full URL is 2083 characters or longer.
- **URL Preview**: the full URL the call goes to, built from the server's base URL, the relative path and the parameters. Until a listed server is chosen, the preview starts with `http://example.com/`.
- **Request method**: *GET*, *PUT*, *POST* or *PATCH*. A new command starts with *GET*.
- **Content type** and **Body**: the content type of the request, for example `text/plain; charset=utf-8`, and the data sent to the server. Shown only when the method is not *GET*.
- **Parameters**: **Name** and **Value** pairs added to the URL as a query string. A parameter replaces one of the same name in the relative path.
- **Headers**: **Name** and **Value** pairs sent with the request as HTTP headers.

Click **+** next to **Parameters** or **Headers** to add a row, and choose **Delete Parameter** or **Delete Header** from the row's options menu, then click **Delete** in the **Confirm delete** dialog. Every row needs a name, and two rows in the same list cannot share one. Parameters and headers cannot be given newsroom tags in Template Editor.

The command list shows the title, the method and the path, and the timeline shows the same on the command.

Info: A template can hold any number of Generic REST commands, on any trigger. A Generic REST command on *Out* does not count toward the single *OUT* command a template supports, so it can sit next to a switcher command on *OUT*.

Info: There is no **Offset** on a Generic REST command, and **Mix Delay** does not apply to it.

Info: The REST servers are not set up in Template Editor. An administrator sets them up in Viz Mosart, each with the base URL your relative path is added to.

Info: If a command's settings cannot be read, the panel says so instead of showing the fields. The command is kept exactly as it is. Edit it in the desktop Template Editor.

Note: Choose a server in **Assign to device**. A command with no server is saved, but the call has nowhere to go. The command list flags it with an alert icon, and hovering over the icon shows why (*No REST server set*).

Copyright © 2026 Vizrt

11.11.9 Fine-tuning On-Air Action Timings

In the Template Editor, you can control **when** certain actions are executed relative to the moment a template is taken on-air. This is done using either:

- A *global* **Mix Delay**, defined in the template properties.
- An *individual* **Offset**, defined per command.

Some commands support both, depending on their configuration.

Mix Delay introduces a shared delay (in frames) applied to eligible commands when the template is taken on-air. It's useful when you want multiple commands delayed together, in a coordinated way.
In addition to a template's Mix Delay, some commands also support their own **Offset** property, an individual delay in frames.

Delay mechanism summary

Command Type	Mix Delay applies if	Offset applies if
Switch Main	Trigger is *TAKE*	Mix Delay is not set
Switch Region	Trigger is *TAKE*	Mix Delay is not set
Set Source	Layer is *Background* and Trigger is *TAKE*	Trigger is *CUE*, or Trigger is *TAKE* with a Layer other than *Background*, unless the command was created in the desktop Template Editor
Set Fader	Fade up trigger is *TAKE*	Not supported
Control Audio Input	Never	Not supported
Macro Recall	Never	Not supported
Camera Shot	Never	Trigger is *TAKE*, on NDI PTZ and Vinten only
Light Control	Never	Not supported

Copyright © 2026 Vizrt

Command Type	Mix Delay applies if	Offset applies if
GPI/O Control	Never	Not supported
Weather	Never	Not supported
Graphics	Never	Not supported
Generic REST	Never	Not supported

11.11.10 Template Terms and Definitions

Some common terminology used in template and show design:

- **Directtakes:** A special template set, where templates in this set are intended to be manually executed in an ad-hoc fashion.
- **Empty template:** An undefined, non-operative template. An empty template does nothing when executed. Note that empty templates are *not* displayed in the Viz Mosart client.
- **Gallery:** Corresponds to a Viz Mosart installation. Typically a gallery has a dedicated Viz Mosart server, with configured utilities and a Viz Mosart client.
- **Show design:** The background task of creating a set of templates, in a logical and programmatically valid order, for controlling a scheduled, rundown-driven, studio transmission.
- **Template description:** The template properties common for all Viz Mosart installations in a shared template set. This description uniquely identifies and describes the template.
- **Template implementation:** Contains the behavior or device control aspects of the template. The template implementation is related to the Viz Mosart installation.
- **Template set:** Viz Mosart templates are always contained in a template set. There may exist multiple *versions* of the same template but in *different* template sets.
  A template is uniquely defined by its *type* and *variant name* within its corresponding template set.
- **Template type:** Template type or category. Viz Mosart has a predefined set of template types. Each are displayed with a distinct visual appearance.
  Examples of template types are CAMERA, PACKAGE, VOICEOVER, LIVE, GRAPHIC and DVE.
- **Template variant:** Unique identifier of a template for given template type within a template set.
  For example, *camera1* and *camera2* are two variants of a Camera template.

Copyright © 2026 Vizrt

11.12 Transitioning from the old Template Editor

Here we review considerations when adopting the new Template Editor.

- Mapping Desktop Template Editor Commands to the Web Apps Template Editor
- Currently Unsupported Command Properties
- Examples
- Legacy commands

11.12.1 Mapping Desktop Template Editor Commands to the Web Apps Template Editor

Desktop TE	Web Apps TE	Notes
Switcher crosspoint	Switch Main Switch Region	*Switching sources* is supported for a maximum of *two* regions in parallel, one of which must be the main output (PP). See Currently Unsupported Properties.
KeyFill	Set Source *Layer*=1, 2, 3 etc. *Trigger*=TAKE	See Currently Unsupported Properties.
Auxiliary	Set Source *Region*=Auxiliary *Layer*=1,2,3 etc. *Trigger*=TAKE	See Currently Unsupported Properties.
SET_CROSSPOINT	Set Source *Layer*=1, 2, 3 etc. *Trigger*=CUE/OUT Set Source *Region*=Auxiliary *Layer*=1,2,3 etc. *Trigger*=CUE/OUT Set Source *Layer*=Background *Channel*=A/B *Trigger*=CUE/TAKE/OUT	See Currently Unsupported Properties.

Copyright © 2026 Vizrt

Desktop TE	Web Apps TE	Notes
SET_AUXILIARY_CROSSPOINT	Set Source *Region*=Auxiliary *Layer*=1,2,3 etc. *Trigger*=TAKE	This control command is visible in the desktop Template Editor but it is not executed by Viz Mosart. This command cannot be recreated in the web-based Template Editor, but a **Set Source** command can be used to assign a source on an auxiliary. See Currently Unsupported Properties.
	Set Source *Region*=Auxiliary *Layer*=1,2,3 etc. *Trigger*=CUE/OUT	Not supported in this version.
Audio Fader	Set Fader	See Currently Unsupported Properties.

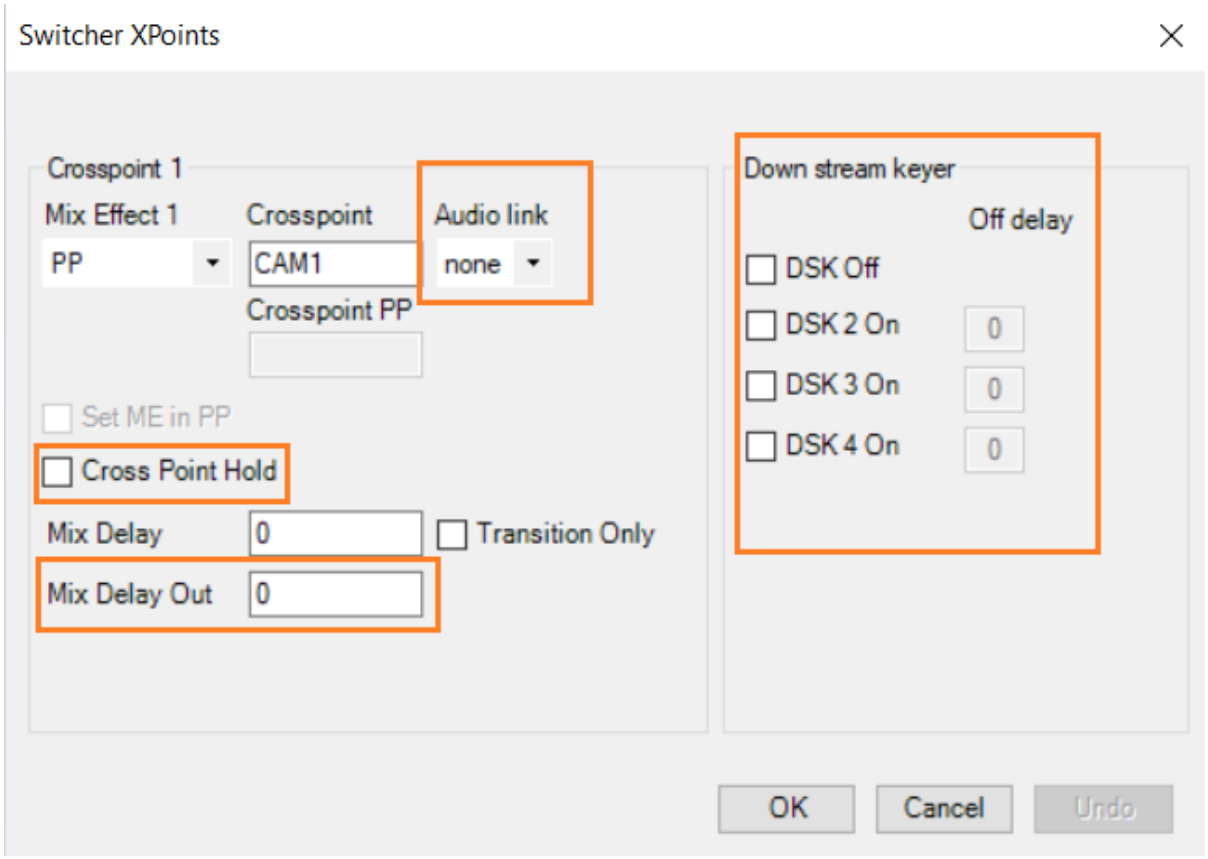
11.12.2 Currently Unsupported Command Properties

This section lists command properties that are currently not supported in the new web-based Template Editor. These properties apply to commands already available in the editor but are either not implemented yet, planned for future support, or will be deprecated or handled differently in upcoming versions.

Switcher Crosspoint

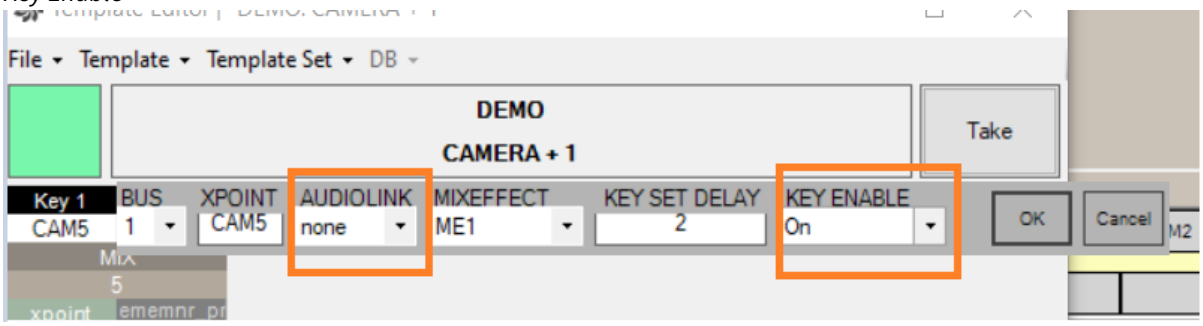
- *Audio link*
- *Cross Point Hold*
- *Down stream keyer*
- *Mix Delay Out*
- The combination *Mix Delay* set to *0* frames and *Transition Only* checked. This executes the video transition with no delay after all the commands have been executed.

Copyright © 2026 Vizrt



KeyFill

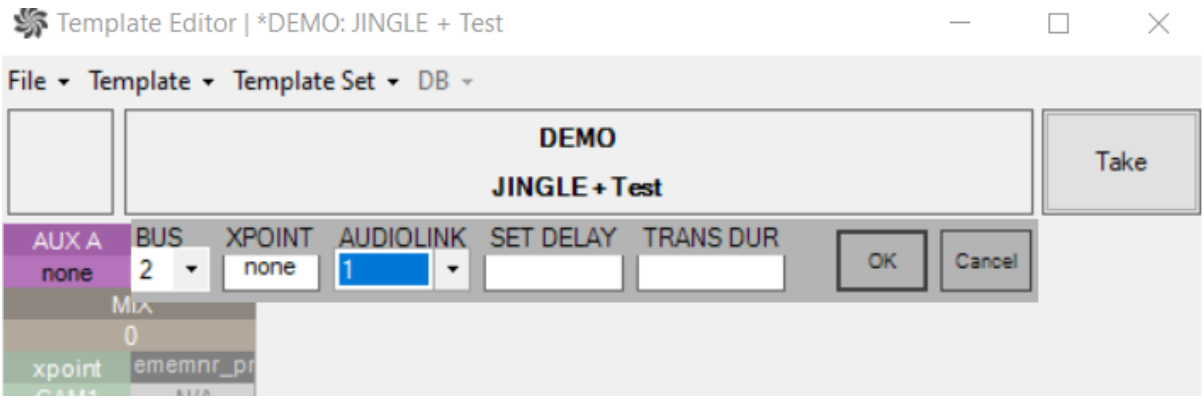
- *Audio link*
- *Key Enable*



Auxiliaries

- *Audio link*
- *Trans Dur*

Copyright © 2026 Vizrt

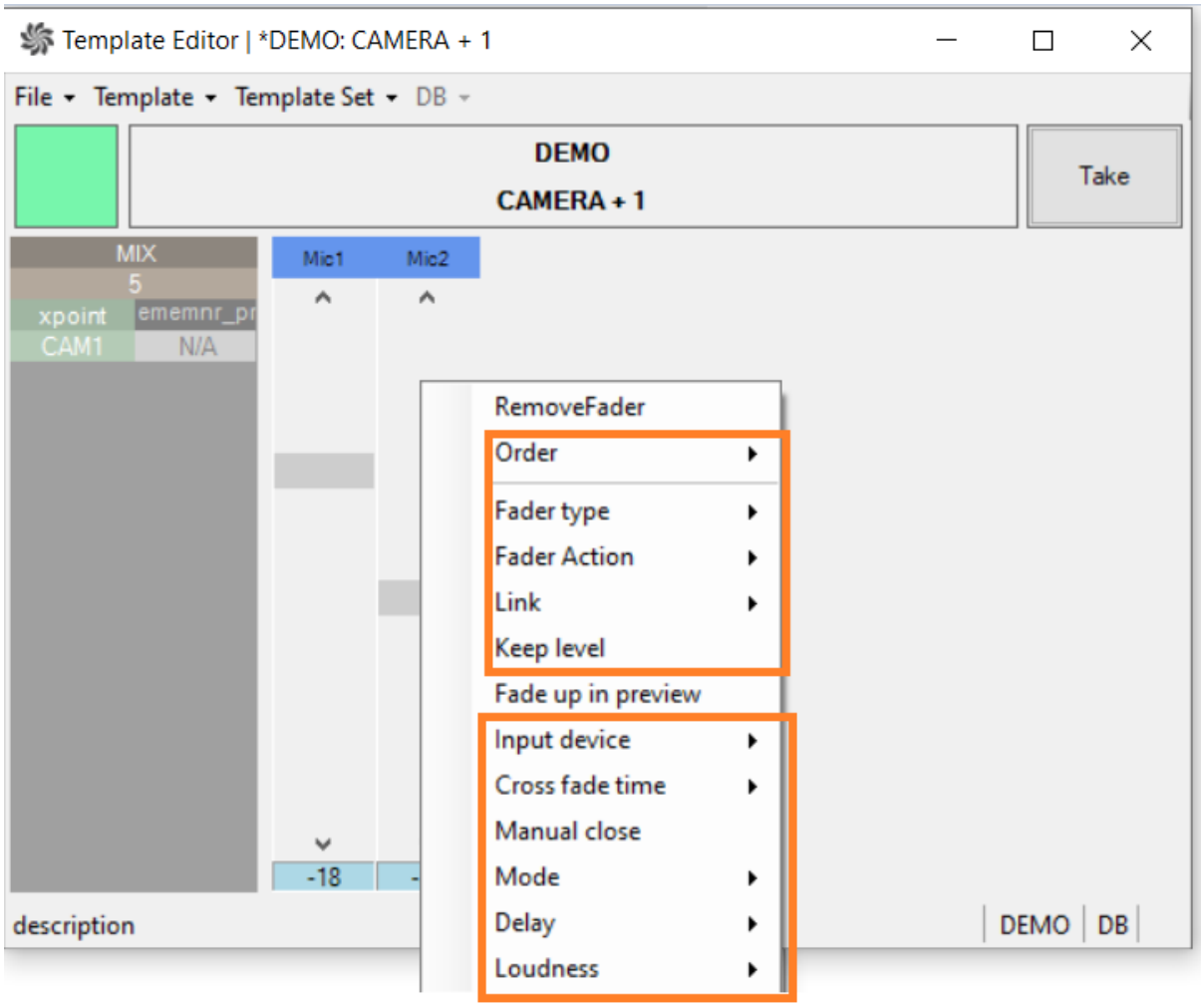


Faders

Most of the fader properties are not yet supported in this version:

- *Order*
- *Fader type*
- *Fader Action*
- *Link*
- *Keep level*
- *Input device*
- *Cross fade time*
- *Manual close*
- *Mode*
- *Delay*
- *Loudness*
- *Set level 2*
- *Set level 3*

Copyright © 2026 Vizrt

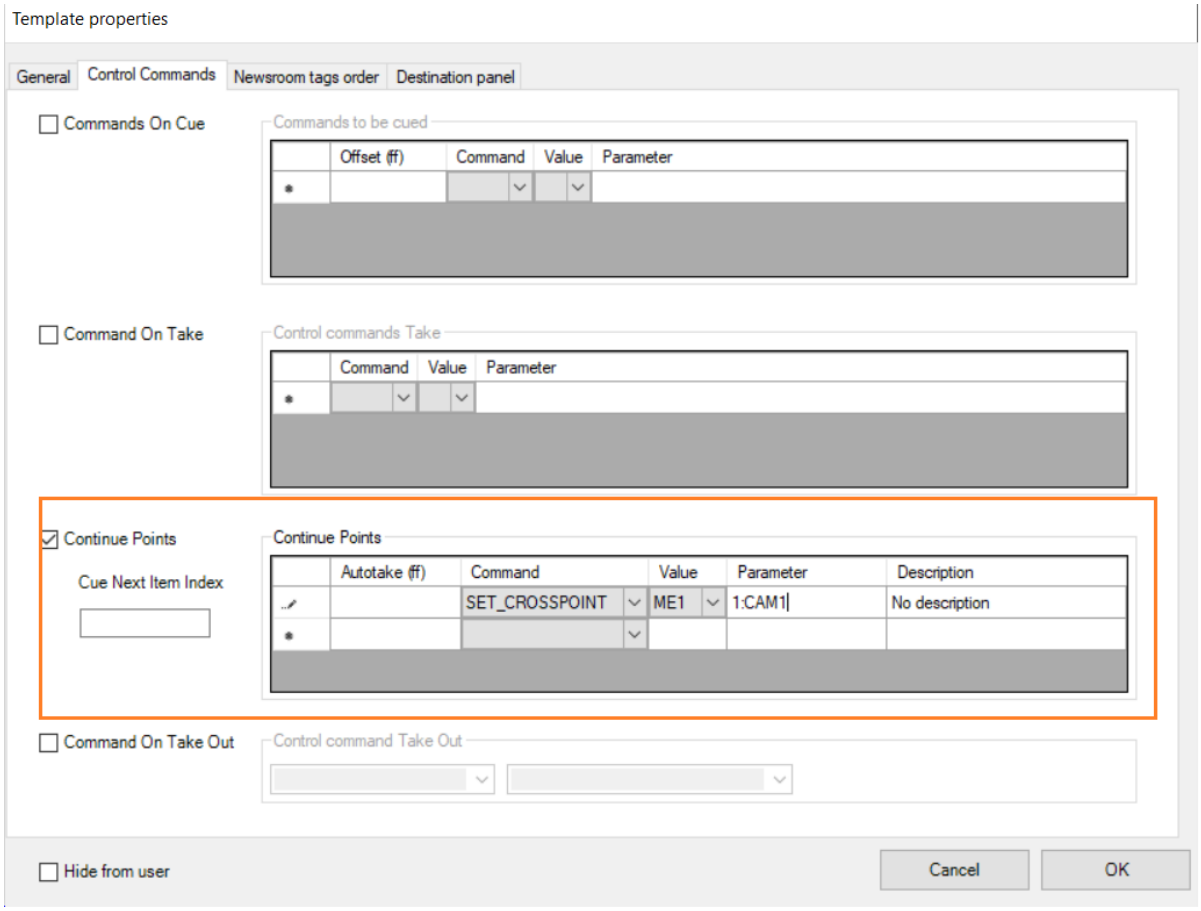


Control Commands

Only SET_CROSSPOINT commands are currently supported in the Web Apps Template Editor with some exceptions:

- SET_CROSSPOINT as a continue point:

Copyright © 2026 Vizrt



Info:

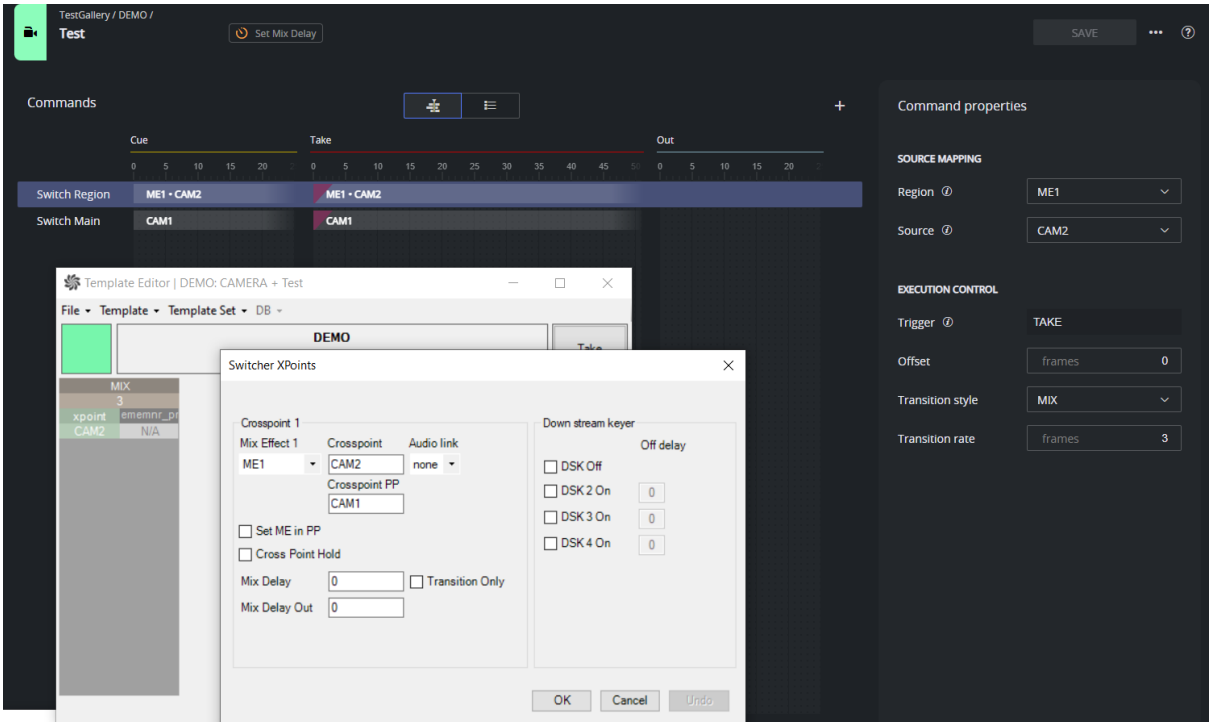
- **SET_CROSSPOINT** with **Value**=*AUX* can be configured in the desktop Template Editor, but it is not actually executed by Viz Mosart.
- **SET_AUX_CROSSPOINT**, although available in the desktop Template Editor, is not executed by Viz Mosart.
- **SET_CROSSPOINT on TAKE OUT** cannot be properly configured in the desktop Template Editor. This *is* however, supported in the new web-based version.

11.12.3 Examples

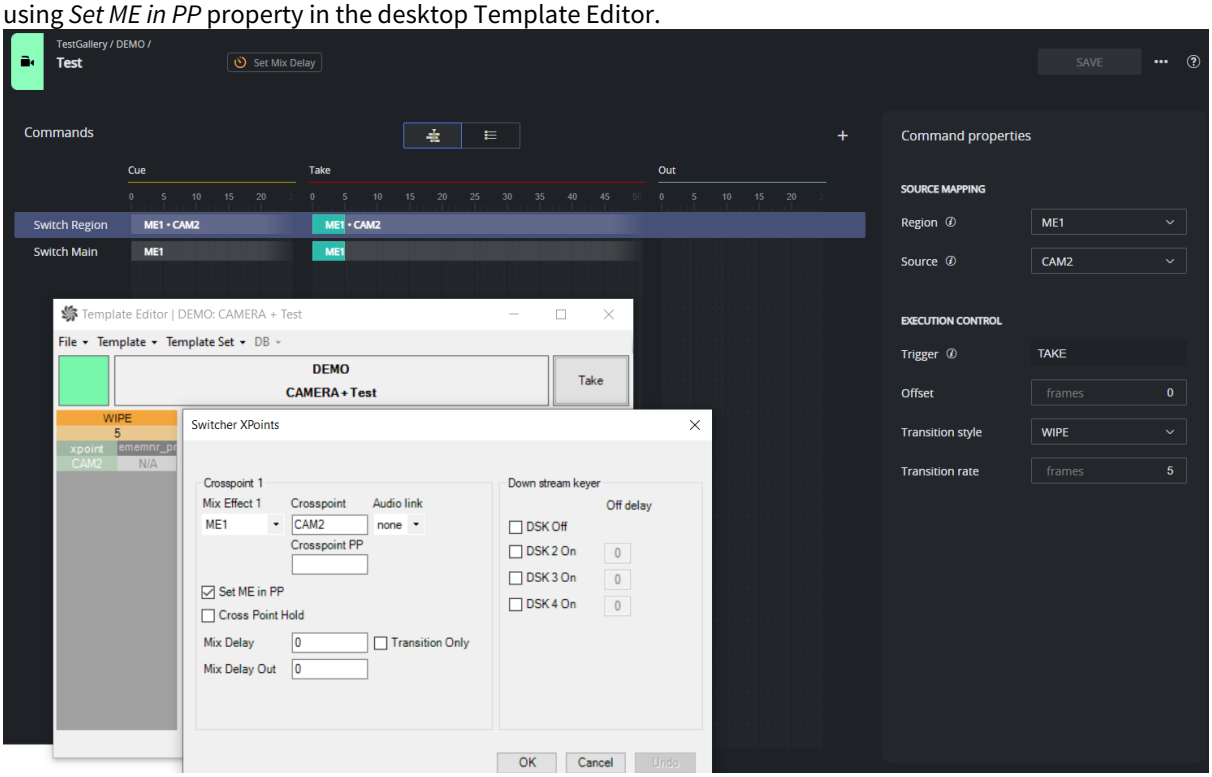
Here are some examples of templates created in the desktop Template Editor and how they are represented in the new Template Editor.

1. Switch different sources on the main output (PP) and another mixer region (ME1) with a mix rate > 0.

Copyright © 2026 Vizrt

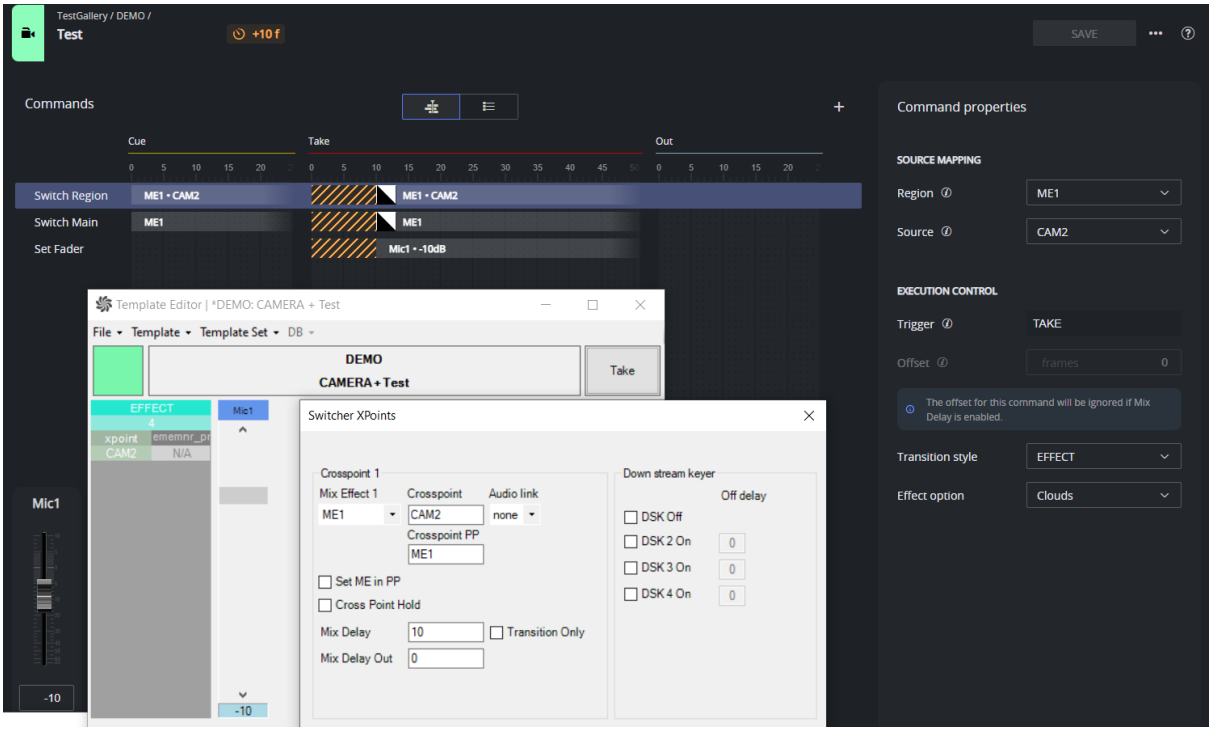


2. Switch M/E into the main output (PP) with a WIPE and rate > 0.
   Note: The same commands could be executed in the web-based if *Crosspoint PP* is set to ME1 instead of using *Set ME in PP* property in the desktop Template Editor.

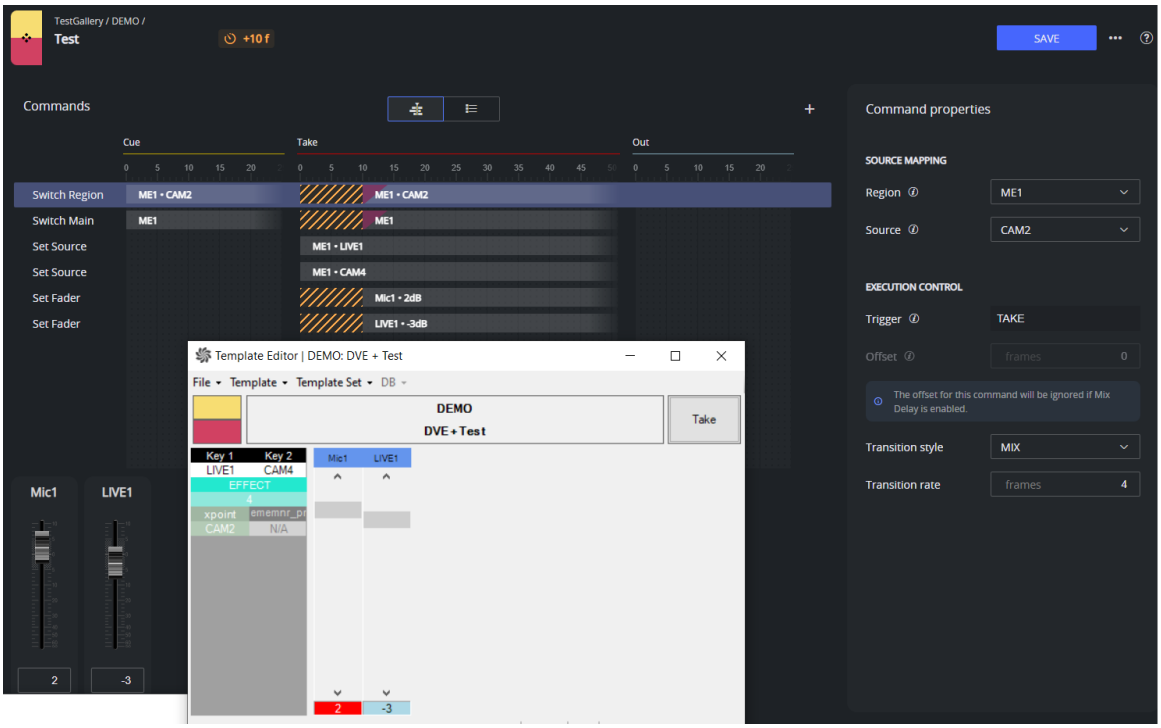


3. Switch a Mix/Effect (M/E) source into the main output using an effect transition, while also applying a mix delay across multiple commands.
   In addition, an audio fader adjustment is performed as part of the same template.

Copyright © 2026 Vizrt

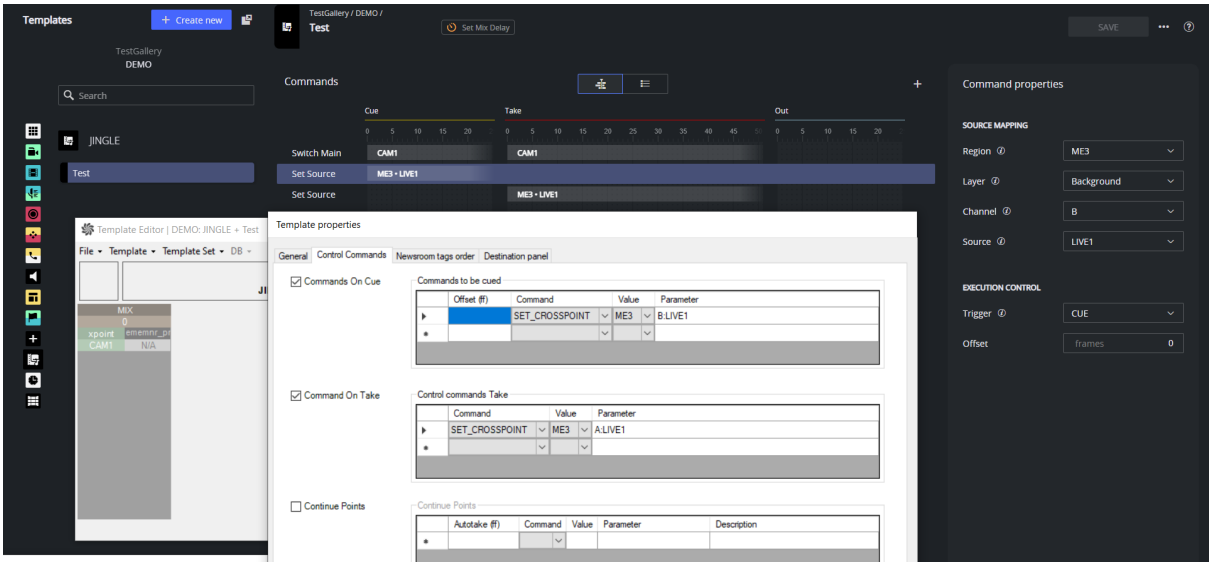


4. Take a DVE composition with multiple video and audio elements.
   In this template, two video sources (LIVE1 and CAM4) are inserted into a DVE layout (via ME1), which is then switched into the main output using a mix transition with a custom rate. Simultaneously, two audio faders (LIVE1 and MIC1) are adjusted. A mix delay is applied across multiple commands to synchronize the execution.

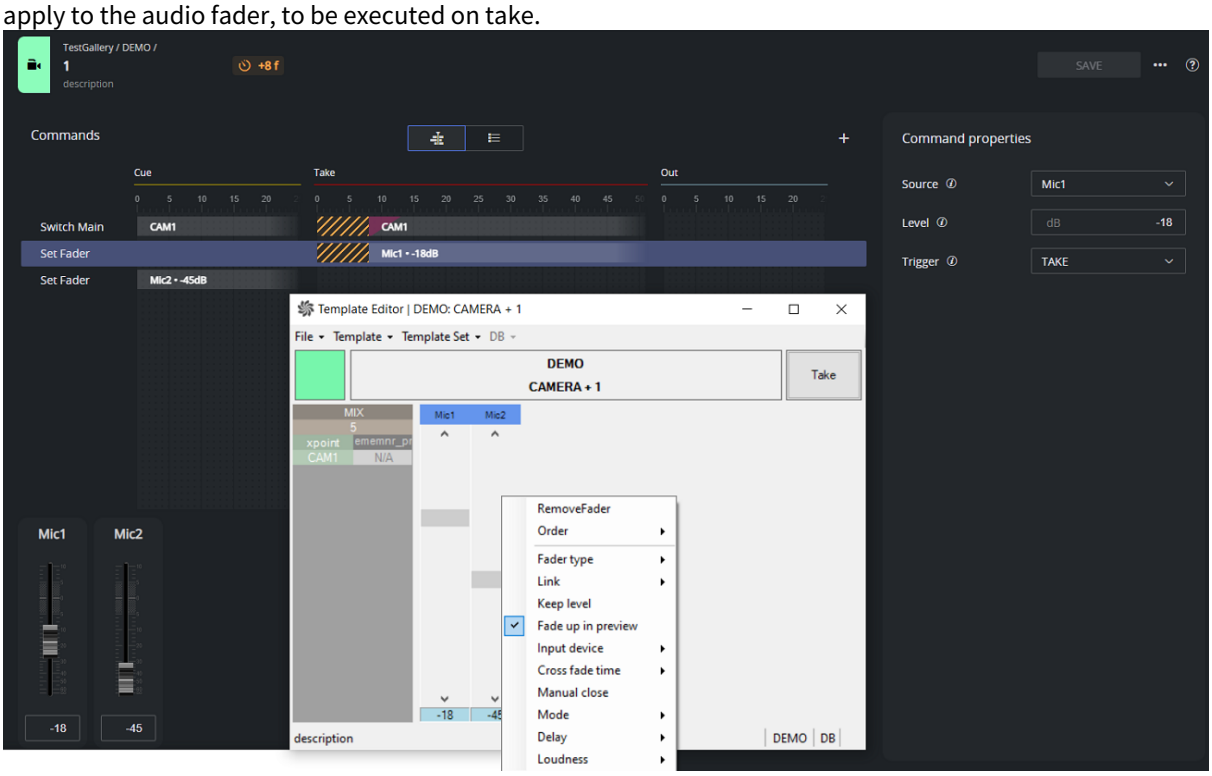


5. Assign a source to ME3 bus B at cue, and transition the same source to bus A on take:

Copyright © 2026 Vizrt



6. Set fader level when the template is cued and when it is taken. A Mix Delay set on the *Switcher crosspoint* will apply to the audio fader, to be executed on take.



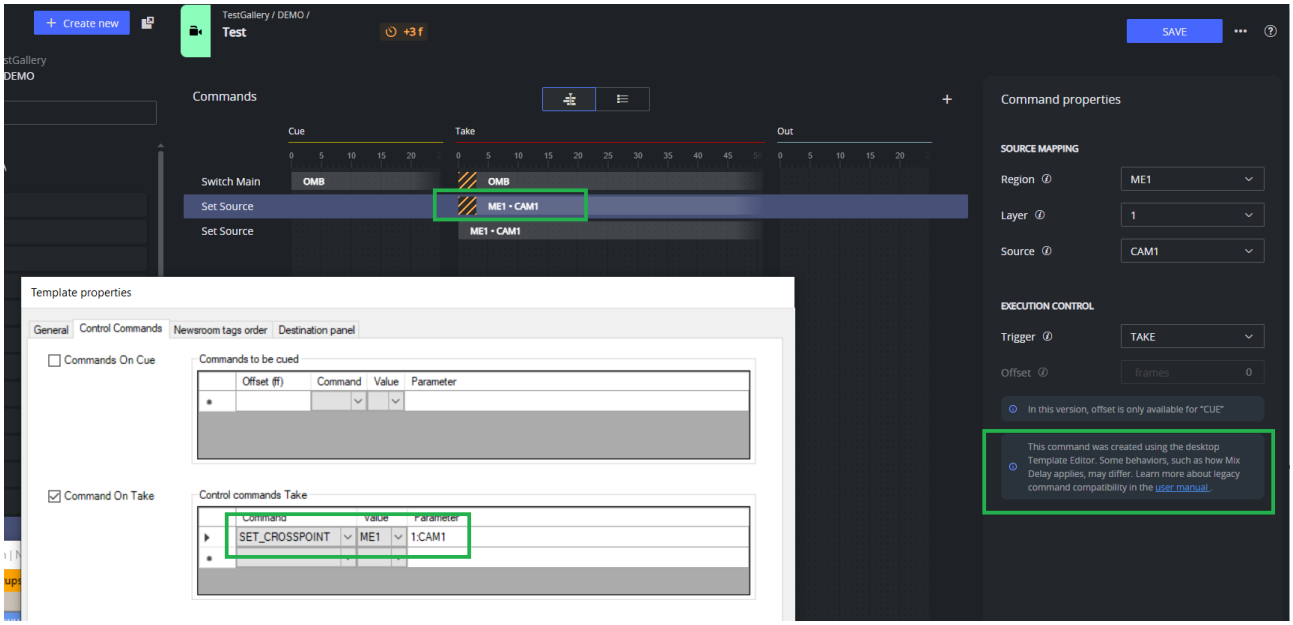
11.12.4 Legacy commands

Control Commands to set the Source on a Mixer Region Keyer

In the desktop Template Editor, you could add a *control command on take* like this:

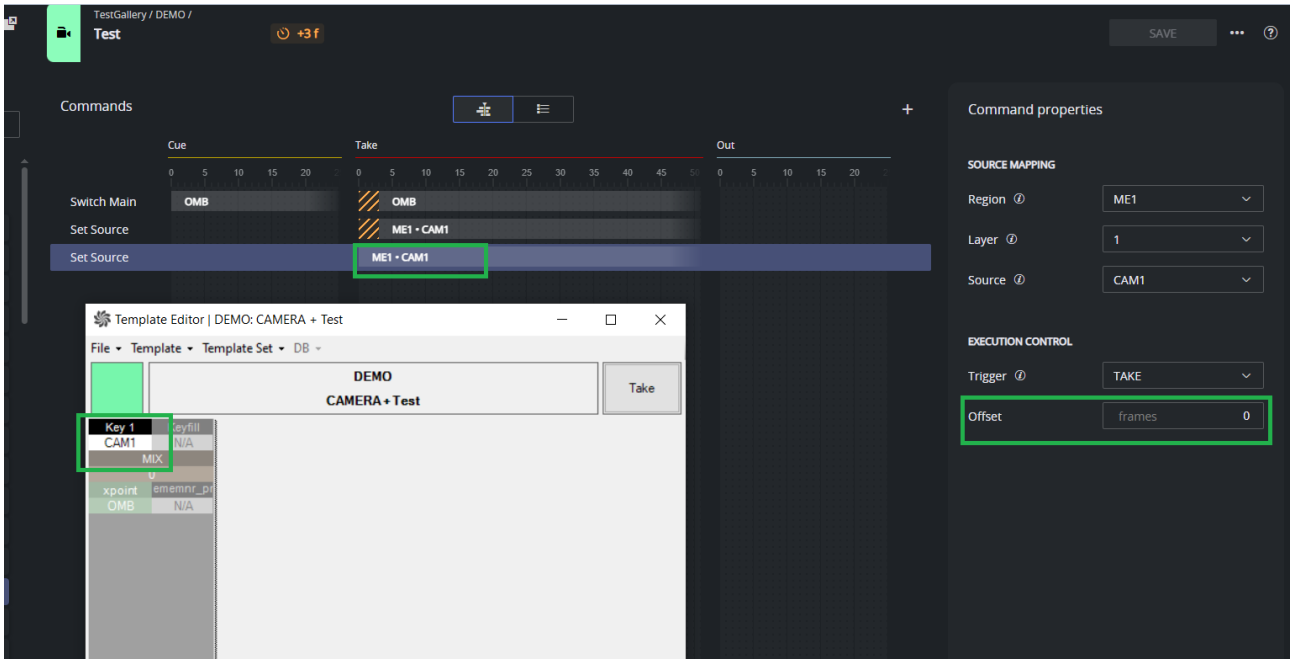
SET_CROSSPOINT on take ME1 "1:CAM1"
This instructs Mosart to switch ME1 to input CAM1, using the bus number 1 to specify the source.

When this type of command is imported into the web-based Template Editor, it is recognized as a *legacy command* and preserved in its original form. If a Mix Delay is set on the template, this applies to *all* the control commands on take. This is visualized in the web-based Template Editor as in the screenshot below:



However, if you recreate a command in the web-based editor with the *same Region and Source settings*, the command will be exported to the desktop Template Editor as a **KeyFill** command. ***Mix Delay*** *will not apply* to this command. Instead, you can use the **Offset** property to delay the command execution relative to the Take.

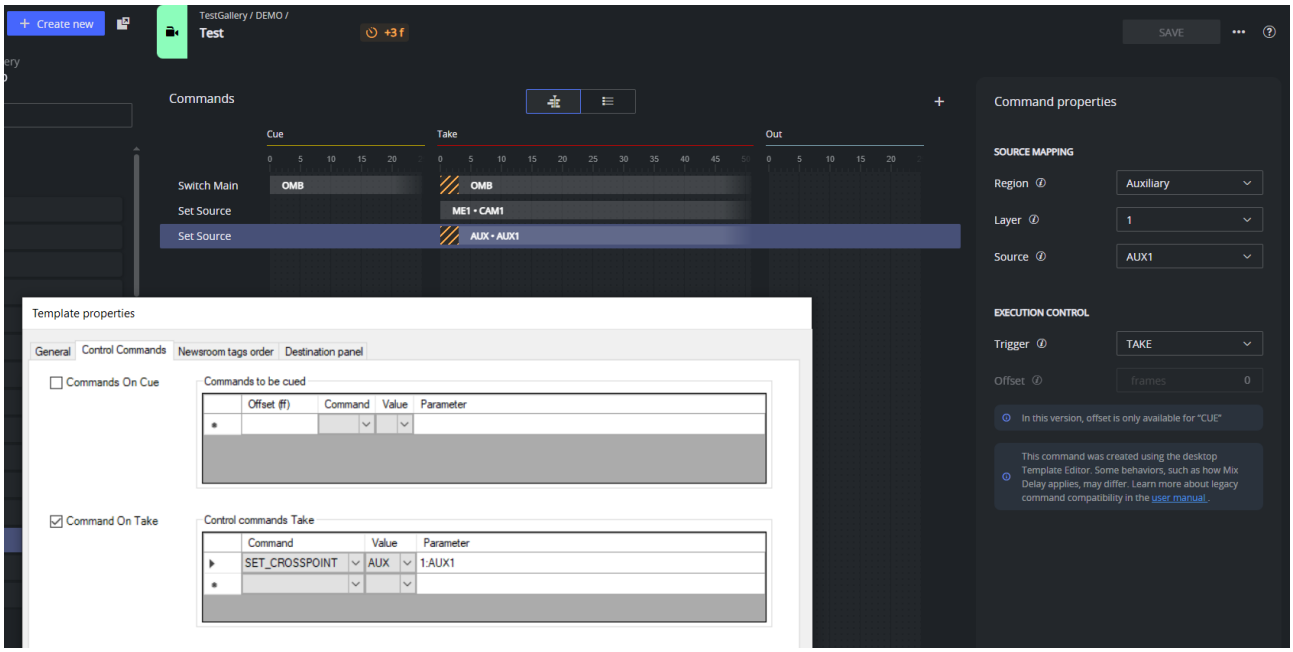
Copyright © 2026 Vizrt



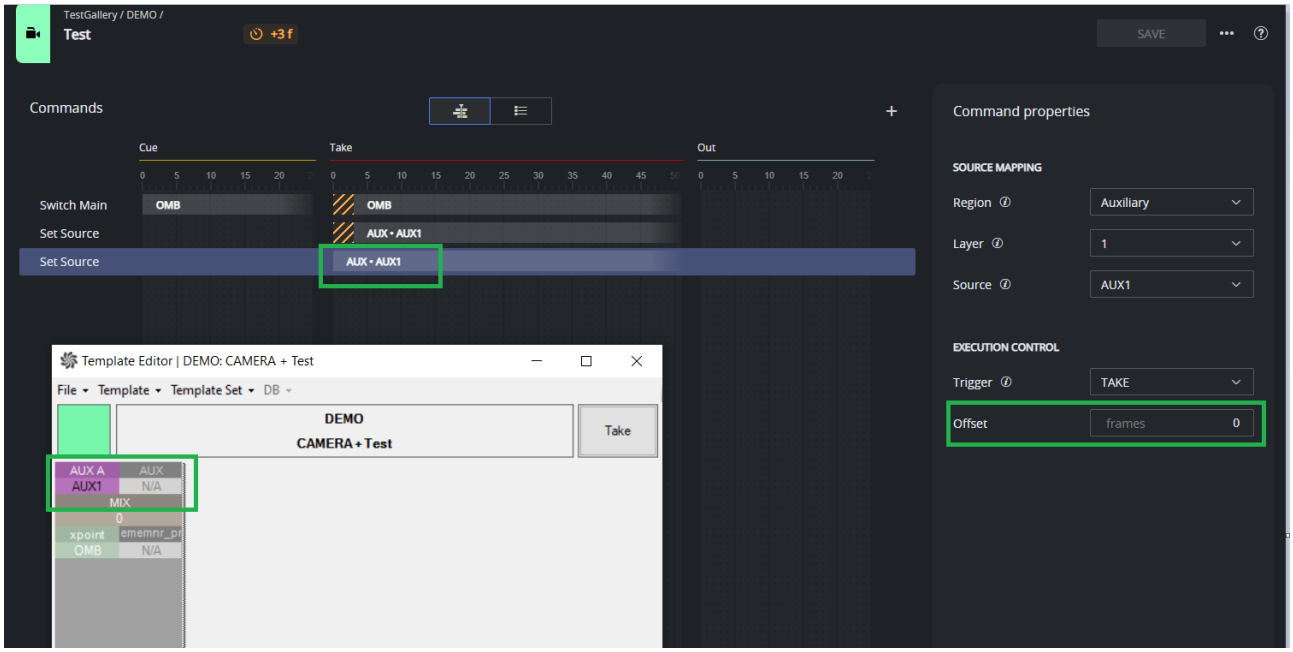
Control Commands to set the Source on an Auxiliary

Control commands to set the source on an auxiliary are also considered *legacy* commands and can not be recreated using the web-based Template Editor.

If a Mix Delay is set on the template, this applies to *all* the control commands on take.
This is visualized in the web-based Template Editor as in the screenshot below:



If a command with exact properties is added in the web-based editor, it is translated into an *Auxiliary* command in the desktop version. ***Mix Delay*** *will not apply* to this command. Instead, you can use the **Offset** property to delay the command execution relative to the Take.



Copyright © 2026 Vizrt

12 Prompter App

The Prompter is a browser-based teleprompter that displays the script from the live Viz Mosart rundown on any connected device: a laptop, a tablet, or a dedicated prompter screen. It needs no additional hardware or software beyond a web browser.

- Follows the on-air rundown automatically: as items are taken on air, the script rolls to the current item and reads ahead across items that have no script of their own.
- A reading cue band and cue arrows mark the line being read, at an adjustable position on the glass. The band can be switched off independently of the arrows.
- On-screen controls and keyboard shortcuts for scroll speed, text size, and manual transport.
- Mirrored output (horizontal and vertical) for teleprompter glass rigs.
- Can be played and paused from a Smart Panel button, and embedded in a Smart Panel as a live widget.
- A display-only mode that shows the rolling script alone, for passive output screens.
- Where authentication is enabled, it holds its sign-in for as long as its page is open, so an unattended prompter is never signed out for inactivity.

12.1 Typical Usage

The Prompter is aimed at presenters, anchors, and prompter operators. In a typical setup, the Prompter runs full screen on the studio prompter display (or on the talent's tablet or laptop) while Viz Mosart drives the show. The presenter reads from the cue line; the script rolls automatically as the automation takes each item, so nobody has to drive the prompter by hand. For rehearsal, or whenever the operator prefers to control the pace, the manual transport controls are available at all times.

12.2 Prerequisites

- Viz Mosart 5.1.0 or later.
- The Prompter installs as part of the **Mosart Web Applications** bundle. No separate installation is required.
- It is opened from **Home** in the Mosart Web Apps Configuration Tool.

This section describes:

- *Working with the prompter* describes the prompter view, following the on-air rundown, the on-screen controls and settings, the reading cue, remote control from Smart Panel, and how its session behaves.
- *Setup and administration of Prompter* covers installation, opening the app, the connection to the Viz Mosart server, the session timeout controls, stored settings, and troubleshooting.

Copyright © 2026 Vizrt

12.3 Working with the Prompter

The Prompter presents the script of the current Viz Mosart rundown as classic teleprompter text: large white type on a black background, rolling at a steady reading pace.

Info: To read a script in the Prompter, a standard Viz Mosart setup must first be running with a loaded rundown.

12.3.1 Opening the Prompter

1. From the Mosart Web Apps Configuration Tool, select **Home** and then the **Prompter** card.
2. The Prompter opens in your default web browser. On a system configured with multiple studios, it opens for the selected studio.

Until a script is available, the Prompter shows one of the following messages in the reading area:

Message	Meaning
Connecting to Mosart…	The Prompter is establishing its connection to the Viz Mosart server.
Waiting for the Mosart Server	The connection is up, but no live timing is arriving from Viz Mosart. Check that Manus Administrator is running.
No rundown loaded	Load a rundown in Viz Mosart and it appears in the Prompter automatically.

A connection chip in the top right corner of the screen shows the live connection state at all times: **Connecting…**, **No Mosart Server**, or **Connected**.

12.3.2 The Prompter View

The whole rundown renders as one continuous script, in running order. What the reading area shows depends on whether the Prompter is following the live item or being browsed.

The view keeps itself up to date: stories and script added, edited, deleted or reordered elsewhere (in Showmaker or in the newsroom system) appear in the Prompter automatically, without reloading the page.

While Following the Live Item

During normal on-air operation the reading area is kept as clear as possible, so nothing but script passes through the cue line:

- The in-flow story headers and item labels fade out, so they never roll up through the reading area and distract the talent.
- Instead, a fixed banner at the top of the screen carries that context: the on-air story and item, so the presenter always knows what is live. Items with nothing to read are tagged **No script** in the banner.

Copyright © 2026 Vizrt

- The state strip stays visible in the top left corner, with the on-air timers just below it. Both are described below.

While Browsing

The moment the operator scrolls or steps away from the on-air item, the full structure of the rundown reappears:

- Every story appears as a numbered header showing its name, and its planned duration when one is set. The on-air story's header also shows a live countdown to the next story.
- Each item in a story is labelled with its Mosart template (for example, a camera or package name) in the template type color, together with the item's planned duration when one is set. Items with script show their script below the label; items with no script show the label only.

In both states, a thin progress bar along the bottom edge of the screen shows the reading position within the whole rundown, and the name of the loaded rundown appears in the top right corner while the control bar is visible.

Story headers of rundowns created in Showmaker also offer an **Edit in Showmaker** button, which opens that story in Showmaker in a new browser tab. Like the control bar, it is only shown while the controls are visible, so it disappears once they auto-hide.

State Strip

Once a rundown is loaded, a compact state strip stays visible in the top left corner of the screen and shows two words. It is deliberately exempt from the control bar auto-hide, so the operator can always read the transport and follow state without touching anything. It is hidden in the connecting and empty states, where a transport word would be misleading, and in display-only mode.

State	Meaning
RUNNING	The script is rolling.
WAITING	The roll is running but held at the end of the on-air item's script, waiting for the next take. Shown only while Follow the rundown is on.
PAUSED	The script is not rolling.
HOLD	A rundown is loaded but the Viz Mosart timeline is not running (a programme hold).
FOLLOWING	The Prompter is tracking the on-air item.
PREVIEW	Follow is on, but the operator has scrolled away from the on-air item. Highlighted as a warning; a **Jump to live** button is offered.
MANUAL	Follow the rundown is switched off; the operator drives the scroll. Highlighted as a warning.

Copyright © 2026 Vizrt

Timers

While **Follow the rundown** is on and the on-air item is found in the loaded rundown, an elapsed timer sits below the state strip in the top left corner, counting up to show how long the on-air item has been live.

A second timer joins it only while an item with **no script** is on air. It counts down to the moment the next scripted item is due to roll, adding together the time left on the on-air item and the planned durations of any further no-script items ahead of it, so a run of back-to-back packages reads as one shrinking number. If the on-air item runs long, the countdown crosses zero and turns red to show that the next script is overdue.

12.3.3 Following the On-Air Rundown

With **Follow the rundown** switched on (the default), the Prompter follows the production automatically, item by item:

- When a rundown is taken on air, scrolling starts automatically; nobody has to press play.
- Taking an item that has script brings that item's script to the cue line and rolls it at the reading pace. The roll holds when the last line of the item's script reaches the cue line, so the Prompter never rolls ahead of what is on air. While it holds, the state strip reads WAITING.
- Taking the same item again (a re-take) re-anchors its script to the cue line and rolls it again from the top.
- A take always restarts the roll, even after a manual pause. A manual pause holds only between takes.

Reading Ahead over Items with No Script

When an item with no script is taken on air (a package, a full-screen graphic, a break, a live source), there is nothing for the talent to read, so the roll stops. What happens next depends on the **Read ahead during no-script items** setting (default on):

- **On**: The Prompter brings the next scripted item in the rundown to the cue line so the presenter can read ahead. The reading cue turns amber, band and arrows alike, to signal that the script on the cue line is not yet on air, and the countdown timer shows when it is expected to roll. The moment that item is taken, it rolls from where it already sits.
- **Off**: The Prompter freezes in place until the next scripted item is taken. The reading cue still turns amber and the countdown timer still shows when the next script is due, so it stays clear that nothing on the cue line is on air.

In both cases the on-air banner keeps showing the live story and item, tagged **No script**.

Browsing While On Air

Scrolling by hand (mouse wheel, touch drag, **Home**, or stepping between items) while *Follow* is on switches the Prompter to **PREVIEW**: the roll stops so the operator can read anywhere in the rundown without being pulled back. To return:

- Select **Jump to live** (or press **L**) to snap back to the on-air item and resume following.
- Or simply wait: the next take automatically returns the Prompter to the live item.

On a touch screen, a stationary tap only reveals the controls. It takes a real drag to switch to PREVIEW, so touching the glass cannot knock the Prompter off air mid-read.

Copyright © 2026 Vizrt

While following, if the on-air item is not present in the loaded rundown (for example it was removed after the rundown was loaded), the Prompter stops the roll and shows the warning *On-air item is not in the rundown* rather than rolling stale script.

12.3.4 The Control Bar

A floating control bar at the bottom of the screen keeps the essential controls within reach. It hides after four seconds without input and reappears on any mouse movement, touch, or Prompter shortcut key. It also stays put while the settings panel or the Help menu is open.

Control	Description
Settings	The gear button opens the settings panel (see below).
Scroll speed	A slider from 0.25× to 6× with a numeric readout. It moves in quarter steps up to 2× and in half steps from 2× to 6×, so the everyday range keeps fine control. The speed is a constant reading pace: the roll advances relative to the text size, so the talent reads the same number of lines per minute at any text size. At the default text size, 1× approximates a typical news reading pace.
Start/pause scrolling	The round button in the center of the bar starts and pauses the roll (also the **Space** key).
Text size	A slider from 24 pt to 96 pt in 3 pt steps, with a readout in points.
Help	Opens a menu offering **Keyboard shortcuts**, **Give feedback** and **About**. The **Documentation** and **What's new** entries are listed but not yet selectable.

The speed and text size controls, the play button and the reading cue are all hidden while one of the messages above is shown, since there is no script to drive.

12.3.5 Settings

The gear button on the control bar opens the full settings panel. Every setting takes effect immediately and is remembered by the browser on that machine, so each prompter position keeps its own setup.

Group	Setting	Description
Text & display	**Font**	The typeface for the script text (default Open Sans). The list offers common fonts found on the machine; **Show all installed fonts** expands it to every installed font (requires a browser permission; available in Chromium-based browsers such as Chrome and Edge).

Copyright © 2026 Vizrt

Group	Setting	Description
Text & display	Text size	The script text size, 24 pt to 96 pt in 3 pt steps (default 42 pt). The same setting as on the control bar.
Text & display	Margins	Extra horizontal margins around the script, 0 to 20 percent of the screen width (default 3 percent).
Text & display	Mirror horizontally	Flips the output left-to-right for teleprompter glass rigs.
Text & display	Mirror vertically	Flips the output top-to-bottom.
Scrolling	Speed	The reading pace, 0.25× to 6× (default 1×): quarter steps up to 2×, then half steps up to 6×. The same setting as on the control bar.
Scrolling	Follow the rundown	Tracks the on-air item and rolls its script automatically (default on). Off hands full manual control to the operator.
Scrolling	Read ahead during no-script items	While an item with no script is on air, brings the next scripted item to the cue line so the talent can read ahead (default on). Shown only while Follow the rundown is on.
Cue indicator	Off / Left / Right / Both	Where the cue arrows appear: on the left edge, the right edge, both edges, or not at all (default Both). Off hides the entire cue indicator, band included.
Cue indicator	Highlight band	Shows or hides the translucent highlight band across the script, independently of the arrows (default on). Not shown while the cue indicator is Off.
Cue indicator	Position	The vertical position of the cue line, 25 to 75 percent from the top of the screen, in 5 percent steps (default 25 percent).

12.3.6 The Reading Cue

The reading cue marks the line the talent should be reading. It consists of two parts:

- **Cue arrows** at the left and/or right edge of the screen, per the Cue indicator setting.
- An optional translucent **highlight band** across the full width of the script.

When following the rundown, each item taken on air aligns its script to the cue line, and the auto-roll keeps the current text passing through it. The cue's vertical position is adjustable in the settings.

Copyright © 2026 Vizrt

Some talent find the full-width band distracting at large studio text sizes. The **Highlight band** setting turns the band off while keeping the arrows, matching the arrow-only cue of traditional prompters. Setting the cue indicator to **Off** hides everything, band and arrows alike.

While the Prompter is reading ahead over an item with no script, the band and the arrows turn amber as a reminder that the text on the cue line is not yet on air.

12.3.7 Manual Transport

For rehearsal, or whenever the operator wants full control of the pace, switch **Follow the rundown** off (the **F** key). The state strip reads MANUAL and the Prompter no longer reacts to takes; the script moves only when the operator drives it:

- Start and pause the roll with the play button or **Space**.
- Adjust the pace with the speed slider or the **Up** and **Down** arrow keys. Each press moves one step on the same scale as the slider.
- Change the text size with the slider or the **+** and **−** keys.
- Step to the previous or next item with **Page Up** and **Page Down**.
- Scroll freely with the mouse wheel or by touch; press **Home** to return to the top.

The scroll controls also work while Follow is on, and switch the Prompter to PREVIEW until the next take or a Jump to live: the mouse wheel, a touch drag, Home, and stepping between items. Play, pause, speed and text size also work, but they do not switch to PREVIEW: the Prompter keeps following, and the next take restarts the roll.

> **Info**: The Prompter uses the operating system's *Reduce Motion* accessibility setting. With it on, jumps between items are made instantly instead of being smooth-scrolled. The reading roll itself is unaffected.

12.3.8 Keyboard Shortcuts

Press **K** (or select **Help > Keyboard shortcuts**) to show this list in the app.

Command	Keys
Start / pause scrolling	Space
Faster / slower	↑ / ↓
Larger / smaller text	+ / −
Follow the rundown	F
Jump to live	L
Previous / next item	Page Up / Page Down
Mirror horizontally	H

Copyright © 2026 Vizrt

Command	Keys
Mirror vertically	V
Cue indicator	C
Scroll to top	Home
Keyboard shortcuts	K

The plus and minus keys work on both the main keyboard and the numeric keypad. Shortcuts do not fire while the keyboard focus is inside a control, a menu or a dialog, so the focused control keeps its own key handling; and the transport keys stay inactive until a rundown is loaded.

12.3.9 Remote Control from Smart Panel

The Prompter can be driven from a Smart Panel:

- A panel button with the **Prompter Play/Pause** action starts and pauses the roll. The play/pause state is shared, so every open Prompter window starts and stops together.
- The **Prompter** widget embeds the live prompter output in a panel, script only with no controls, for confidence monitoring alongside other panel items.

Display preferences (font, text size, mirroring, the cue indicator) remain local to each Prompter window and are not affected by remote control.

Tip: To add buttons and widgets to a panel, see the *Smart Panel* section of this guide.

12.3.10 Session and Sign-in

This applies only where authentication (OIDC) is enabled for Mosart Web Apps. With authentication off, the Prompter never asks anyone to sign in.

The Prompter is an on-air surface, so it holds its sign-in for as long as its page is open. It is never signed out for inactivity, and nobody has to touch the glass to keep it alive. A prompter left running unattended through a show stays signed in.

The one case where a session can still end is one that could not be held: the Mosart Web Apps host was unreachable for a long stretch, or the machine was switched off, or the page was reloaded after the session had already gone. The Prompter then shows a **Session timeout!** dialog with a **Login** button, and script stops updating until somebody signs in again.

Note: The session dialogs are the one thing display-only mode does not hide. Where authentication is enabled, an expired session shows its dialog even on a passive screen or inside an embedded Smart Panel prompter widget, so the display reports that it needs signing in again rather than sitting silently on stale script.

Your administrator can change this and let the Prompter time out when idle, though it is not recommended for an on-air position. See Setup and Administration of Prompter.

Copyright © 2026 Vizrt

12.4 Setup and Administration of Prompter

- Prerequisites
- Installation
- Opening the Prompter
- Connection to the Viz Mosart Server
- Optional Server Setups
- Session Timeout
- Stored Settings
- Remote Control
- Troubleshooting

12.4.1 Prerequisites

- Viz Mosart 5.1.0 or later.
- A running Viz Mosart system. As a minimum, the Viz Mosart Remote Control Service must be reachable, and Manus Administrator must be running with a loaded rundown for a script to appear.
- A modern browser. The latest Chrome or Edge is recommended, as for all Mosart Web Applications.

12.4.2 Installation

The Prompter requires no separate installation and no additional configuration:

- It installs as part of the **Mosart Web Applications** bundle and is listed in the installer's application list.
- It is served from the Mosart Web Apps host at `/prompter` and routed through the Mosart Web Proxy, like the other web applications.

12.4.3 Opening the Prompter

1. Ensure your Viz Mosart system is operating under normal working conditions (Manus Administrator running, a rundown loaded).
2. From the Mosart Web Apps Configuration Tool, select **Home** and then the **Prompter** card.

On a system configured with multiple studios, the Prompter is opened per studio: **Home** opens it against the studio you select, and the app connects to that studio's Viz Mosart server.

12.4.4 Connection to the Viz Mosart Server

The Prompter receives the rundown, the script, and live timing over the same server connection as the other Mosart Web Applications: the Viz Mosart server defined for its studio on the *Studios* page of the Configuration Tool. By default it connects to the Mosart Web Apps host it was served from, which proxies the connection to the configured Viz Mosart server.

For special setups, the connection can be adjusted with URL parameters:

Copyright © 2026 Vizrt

Parameter	Description
`?display`	Display-only mode: shows the rolling script and its reading cue only. The control bar, the state strip, the timers, the on-air banner, the connection chip, the placeholder messages and the *Edit in Showmaker* buttons are all hidden; the thin reading-position progress bar along the bottom edge remains. This is the mode used when the prompter output is embedded in a Smart Panel widget, and it suits any passive output screen. The session dialogs are the one exception to this hiding. Where authentication is enabled, an expired session shows its **Session timeout!** dialog even in display-only mode, so a passive screen or an embedded widget reports that it needs signing in again rather than sitting silently on stale script.
`?signalr=<url>`	Overrides the address of the Viz Mosart server connection (the Mosart SignalR hub), for example `?signalr=http://mosartserver:55167/mosarthub`. Normally not needed; the default routing through the web proxy is recommended.
`?studioId=<id>`	Connects the Prompter to a specific studio on a multi-studio system. Set automatically when the app is opened from **Home** or embedded in a Smart Panel opened for a studio; useful when bookmarking a prompter display for a particular studio.

Note: The two address parameters are not combined. If `?signalr` is given, it wins and `?studioId` is ignored, so use one or the other.

12.4.5 Optional Server Setups

Support for Main/Backup Connection

- If you have a *backup server* for redundancy, define the environment on the *Studios* page of the Configuration Tool, where each studio carries its own main server and optional backup.

Support for HTTPS

- If you are using *HTTPS*, define the environment in the Configuration Tool menu section *Server Configuration*.

12.4.6 Session Timeout

Where authentication (OIDC) is enabled, the controls are on the **Server Configuration** page of the Configuration Tool, inside the **Authentication (OIDC)** block, under **Allow session timeout**. See Authentication for the full description.

- **Prompter** ships with its switch *off*, reading **Stays signed in while open**. The Prompter renews its own session for as long as its page is open, so an unattended prompter display never expires from inactivity.

Copyright © 2026 Vizrt

- **Session timeout** sets the window in minutes, 30 by default. It applies only to the apps whose switch is on.

> ⚠ **Warning**: Do not switch the Prompter's row on for a studio prompter position. Nobody touches a prompter screen during a show, so the session would expire from inactivity while script and timing were still arriving, and the presenter would be reading behind a sign-in dialog. Leave it on **Stays signed in while open** for any on-air surface.

12.4.7 Stored Settings

All Prompter settings (font, text size, margins, mirroring, speed, Follow the rundown, read-ahead, and the cue indicator) are stored locally in the browser on each machine. Nothing is stored on the server:

- Each prompter position keeps its own setup. A studio prompter screen, the director's laptop, and a presenter's tablet can each hold different text sizes and cue positions.
- Settings survive reloads and browser restarts on the same machine and browser profile.
- Clearing the browser's site data resets the Prompter to its defaults.

12.4.8 Remote Control

The Mosart Web Apps host provides a shared prompter control channel used by Smart Panel:

- A Smart Panel button with the **Prompter Play/Pause** action toggles the roll on every connected Prompter. The button calls the Mosart Web Apps host at `/api/v1/prompter-control/toggle`; the host holds the shared play/pause state and pushes it to every Prompter over the `/prompter-control` WebSocket channel, so the current state is also restored whenever a Prompter reconnects.
- The Smart Panel **Prompter** widget embeds the prompter output (display-only mode) in a panel.
- Display preferences remain local to each Prompter window; remote control affects only the transport.

No additional configuration is required; the control channel is part of the standard Mosart Web Applications installation. Note that it is a separate channel from the Viz Mosart server connection: remote play/pause keeps working even while the Prompter shows *Waiting for the Mosart Server*.

12.4.9 Troubleshooting

Symptom	What to check
Connecting to Mosart… does not go away	The Prompter cannot open its connection to the Viz Mosart server. Check that the browser can reach the Mosart Web Apps host, that the Mosart Web Proxy service is running, and that the proxy can reach the Viz Mosart Remote Control Service on the host and port set for the Prompter's studio on the *Studios* page of the Configuration Tool.

Copyright © 2026 Vizrt

Symptom	What to check
Waiting for the Mosart Server, or the connection chip shows *No Mosart Server*	The connection is up, but no live timing has arrived for several seconds. Viz Mosart broadcasts timing about once a second while it is running, so this points at the Viz Mosart side rather than the network. Check that Manus Administrator is running and that the studio's Viz Mosart server host and port are correct on the *Studios* page of the Configuration Tool.
No rundown loaded	The connection is healthy. Load a rundown in Viz Mosart and it appears automatically.
On-air item is not in the rundown	An item is reported on air but is not in the loaded running order, so the Prompter has parked rather than rolling stale script. Typical causes are an item removed after the rundown was loaded, a pre-roll, or a stale on-air reference. The warning clears as soon as an item that is in the rundown is taken. Shown only while **Follow the rundown** is on.
The script does not roll on a take	Check the state strip: it must read FOLLOWING. If it reads MANUAL, switch **Follow the rundown** on; if it reads PREVIEW, select **Jump to live**. Also note that an on-air item with no script does not roll; the Prompter reads ahead or holds, per the read-ahead setting.
A prompter display shows a **Session timeout!** dialog	The sign-in for that browser has ended. The Prompter holds its own session while its page is open, so this points at a long outage between the display and the Mosart Web Apps host, at a machine that was switched off or rebooted, at a page reloaded after the session had gone, or at the Prompter's row having been switched on in **Allow session timeout**. Sign in again, then check that row in **Server Configuration**.
The Prompter card on **Home** shows a warning icon	The Configuration Tool cannot reach the main Viz Mosart Server for the selected studio, so the Prompter (and the Rundown Viewer) may not function properly. The Configuration Tool's status banner gives the reason: either the Viz Mosart Remote Control Service is not running, or the Viz Mosart Server version may be incompatible with this version of Mosart Web Applications. If HTTPS is used, it must be enabled on both the Viz Mosart Server and Mosart Web Applications. Verify *Server Configuration*, then check the web server log files or the proxy server logs, both linked from the Configuration Tool.
The Smart Panel **Prompter Play/Pause** button has no effect	The button reaches the Prompter over a WebSocket channel served by the Mosart Web Apps host, not over the Viz Mosart server connection. Check that the Prompter window can reach the host and that WebSockets are not blocked between them. The transport is all the button controls: if the state strip reads MANUAL, the roll starts but the Prompter still does not follow takes.

For general connectivity issues, please also refer to the *Troubleshooting* section for the Mosart Web Applications.

Copyright © 2026 Vizrt

13 Named Overlays Editor

The **Named Overlays Editor** is a web-based tool for creating and managing the named overlay graphics that Viz Mosart triggers during a live show. Named overlays are pre-configured graphics, such as lower thirds, tickers, breaking news banners and full-screen graphics, that are recalled by name from templates and control commands.

> **Tip**: Refer to the Viz Mosart Administrator Guide, section Overlay Graphics Interface > Named Overlay Graphics.

The editor lets users set up overlay graphics from different sources, configure how they behave on air (timing, destination handler, and when they are taken out), and organize them for use in automation workflows.
It is part of the Mosart Web Apps package and is served alongside the other Mosart web applications.

13.1 Key Features

- **Create, edit and delete overlay graphics**, with field-level guidance and validation.
- **Viz Pilot Edge integration**: browse graphics templates from a Viz Pilot Data Server in an embedded plugin
- **Custom overlays**: create overlays by hand with full control over every field.
- **NamedCG actions**: have Mosart clear all on-air graphics, or take out a specific graphic, whenever an overlay is taken on air.
- **NCS content placeholders**: map newsroom metadata, such as the story slug or item ID, into overlay content fields for substitution at runtime.
- **XML import and export**: import overlays from XML, including the legacy Mosart `NamedOverlayGraphics.xml` format, and export the current set for backup or migration.
- **Search, filter and sort**: find overlays by text, filter by handler or out mode, and sort by name, handler, provider, or the order they were added.
- **Sync to backup server**: in a main/backup setup, one click copies the main server's named overlays to the backup server, so the two stay aligned. Syncing is not automatic in this version, so run it after changing overlays, but what used to be a manual file exercise is now a single button, with a report of what was copied and removed.
- **Mosart server status**: a live connection indicator for the main and backup servers, right in the toolbar.
- **Thumbnail previews** for Pilot Edge graphics, resolved from the Pilot Data Server.

13.2 Typical Workflow

1. **Configure the graphics provider.** An administrator sets up the Viz Pilot Edge plugin. See *Setup and administration* for details. The provider is optional: custom overlays and XML import work without it.
2. **Open the Named Overlays Editor** from Home in the Mosart Web Apps Configuration Tool, or browse directly to `/namedoverlayseditor/` on the Mosart Web Apps host.
3. **Add overlay graphics.** Use the sidebar to open Viz Pilot Edge, browse the available templates, and send one to the editor. Alternatively, create a custom overlay by hand.
4. **Configure overlay properties.** For each overlay, set the destination handler, the take-in delay in frames, and when the graphic is taken off air, which can be after a set duration (Timecode), at the end of the background item, at the end of the story, on an open end, or manually. Optionally add NamedCG actions and NCS content placeholders.

Copyright © 2026 Vizrt

5. **Organize and review.** Use search, filters and sorting to manage the overlay list, adjust descriptions and settings, and delete overlays you no longer need.
6. **Back up or migrate.** Export the overlay set to XML for backup or to move it to another environment. In a main/backup setup, use **Sync to backup server** after your changes so the backup carries the same overlays. Overlays themselves live on the Mosart server and are available to automation as soon as they are created.

Related pages

- Working with NOE: the interface, adding and editing overlays, actions, placeholders, import and export, and keyboard shortcuts.
- Setup and Administration NOE: prerequisites, provider configuration, authentication and roles, main and backup servers, and troubleshooting.

Copyright © 2026 Vizrt

13.3 Working with NOE

You can create and manage Viz Mosart overlay graphics with the Named Overlays Editor.

- Overview
- Session and Sign-in
- The Overlay Table
- Adding Overlays
- Editing an Overlay
- Managing Overlays
- NamedCG Actions
- NCS Content Placeholders
- XML Import and Export
- Keyboard Shortcuts

13.3.1 Overview

The Named Overlays Editor interface consists of the following areas:

- **Toolbar** (top): the search field, three filter dropdowns (handler, out mode, provider), a count of the overlays currently listed, the sort dropdown, the active studio, the Mosart server indicator, and the help button. Where authentication is enabled, the signed-in user's account icon sits at the far right. The active studio is a plain label on a single-studio installation and a dropdown when several studios are configured. Named overlays live on each studio's own Mosart server, so picking another studio reloads the editor and shows that studio's overlays.
- **Sidebar** (left): **Pilot Edge** opens the graphics plugin panel, **Custom** creates an overlay by hand, and below the separator **Import XML** and **Export XML**. In a redundancy setup an additional **Sync to backup server** button appears.
- **Provider panel** (left, resizable): embeds the Viz Pilot Edge plugin as a panel. The panel width can be resized by dragging the edge.
- **Overlay table** (center): one row per overlay, with sortable columns. Before any overlays exist it shows *No overlays added yet*.

Who can edit

- Creating, editing, deleting and importing overlays requires the administrator role.
- Users with the viewer role see the overlay table and can search, filter, sort and export it. The overlay dialog does not open for them: the row buttons are hidden, double-click and Enter do nothing, and the Pilot Edge, Custom and Import XML buttons are not shown in the sidebar.
- When authentication is disabled, everyone is treated as an administrator.
- See *Setup and Administration* for role names.

Server Indicator

At the right of the toolbar, a status dot and label show the Mosart server the editor is working against. The label is the server's own name as reported by Viz Mosart, falling back to *Main* when no name is available. The dot

Copyright © 2026 Vizrt

distinguishes an active server from an idle one and from a server that cannot be reached, and hovering it shows the server version. In a redundancy setup a second label appears for the backup server.

Help

The **Help** button, marked with a question mark at the right of the toolbar, opens a menu with three entries:

- **About**: Application version and build information.
- **Keyboard Shortcuts**: Keyboard shortcut reference, also listed below.
- **Help**: Introduction to named overlays, with a link to *Named Overlay Graphics* in the Viz Mosart Administrator Guide.

13.3.2 Session and Sign-in

This applies only where authentication (OIDC) is enabled for Mosart Web Apps. With authentication off, the editor never asks you to sign in.

The editor signs you out after a period without activity. In earlier versions an editor left open stayed signed in for as long as the page was open, because its own background traffic kept the session alive. That background traffic no longer counts as activity, so an editor abandoned on a shared newsroom machine now expires as it should.

Working in the editor keeps the session alive: typing, clicking, scrolling and moving the pointer all count. The embedded Viz Pilot Edge panel is a separate frame, so the editor cannot see what you do inside it. Having the panel in focus stands in for activity for five minutes after your last action in the editor, and no longer, while a graphic sent back from the plugin counts as activity in full. A long spell spent only inside the panel can therefore still reach the countdown, so click in the editor now and then, or answer the countdown dialog, to hold the session.

Ninety seconds before the session would end, a dialog headed **You will be logged out soon** counts down and offers **Continue session** and **Log out now**. If the countdown runs out you get **Session timeout!** and a **Login** button.

> **Info**: An administrator can put the editor back to holding its session for as long as its page is open, which matters if you leave it running on a spare screen. See Setup and Administration NOE.

13.3.3 The Overlay Table

Each overlay occupies one row. The columns are:

Column	Shows
(first, unlabelled)	Graphic's thumbnail when one is available, and an icon identifying the provider the overlay came from. Hovering the icon names the provider.
Slug	Overlay name, used to recall it from templates and control commands.
Description	A free-text description.
Handler	Destination handler, for example CG or DSK.

Copyright © 2026 Vizrt

Column	Shows
Out on	Graphics Out On mode, for example Timecode or Story end.
In	Take-in delay in frames.
Duration	Duration in frames.
Action	Configured NamedCG action, shown as a small badge, when one is set. Empty when the action is None.
(last, unlabelled)	Row action buttons, that appear on hover.

The **Slug**, **Description**, **Handler**, **Out on**, **In** and **Duration** columns can be sorted by clicking the column header, which is often quicker than the sort dropdown when you are looking for one particular overlay. **In** and **Duration** sort numerically.

Hovering a row reveals three buttons:

- **Edit**: opens the **Edit overlay** dialog.
- **Open graphic**: sends the overlay back to the Pilot Edge plugin for re-editing. It is enabled when the overlay carries a provider element reference, or when it has a graphics ID and a Pilot Edge plugin is configured; otherwise it is greyed out.
- **Delete**: deletes the overlay, after confirmation.

13.3.4 Adding Overlays

There are two ways to add overlay graphics to the editor, plus XML import.

From Viz Pilot Edge

1. Click the **Pilot Edge** button in the sidebar. The Pilot Edge plugin opens in the provider panel.
2. Browse or search for a graphics template in the Pilot Edge interface.
3. Select a template and send it from the Pilot Edge plugin. If an overlay already uses that slug, the dialog does not open: a warning says the overlay already exists, and the editor selects and scrolls to the existing row so you can see which one owns the name. Use **Open graphic** on that row to re-edit its graphic instead.
4. The editor receives the graphic data and opens the overlay dialog with fields already filled in from the template, including the slug, description, graphics ID, MOS ID, handler name, In, Duration and Graphics Out On.
5. Review and adjust the properties as needed, then confirm.

Tip: When importing from Viz Pilot Edge, the overlay's thumbnail is automatically resolved from the Pilot Data Server and shown in the first column of the table.

Custom Overlays

1. Click the **Custom** button in the sidebar, or press **N**. The **Add custom overlay** dialog opens.

Copyright © 2026 Vizrt

2. Fill in the two required fields, which are marked with an asterisk:
    - **Slug**: the unique name that templates and control commands use to recall this overlay, for example `LOWERTHIRD`. It must be unique across all overlays.
    - **Graphics ID**: the graphic's ID in the graphics system, for example the Pilot element ID.
3. Optionally complete the rest of the dialog, which is organized into **General** (Description and MOS ID), **Integration & Timing** (Handler Name, In, Duration and Graphics Out On), **NCS Content**, and **Actions**.
4. Click **Add**.

Every field has an information icon next to its label with a short description of what the field does.

13.3.5 Editing an Overlay

Double-click a row, or select it and press **Enter**, or click the **Edit** button that appears on hover. The **Edit overlay** dialog opens with the same sections as the add dialog. Click **Save** to apply your changes, or **Cancel** to discard them.

General

- **Slug**: the overlay name. It opens read-only, because templates and control commands refer to it. To change it, click the **Rename** button next to the field, type the new name and confirm in the **Rename overlay** step, which warns that templates and control commands still referring to the old name will no longer find this overlay and offers **Keep a copy under the old name**.
- **Description**: free text, shown in the table and in the Overlay Graphics Interface.
- **Graphics ID**: the graphic's ID in the graphics system.
- **MOS ID**: the MOS ID of the graphics device that plays this overlay.

Integration & Timing

- **Handler Name**: the destination handler that plays this graphic, typed as free text. The values Viz Mosart recognizes are CG, DSK, PKSPLIT, LAYOUT, MIXER, VSAR, and CONTENT1 to CONTENT4. An overlay saved with the field empty is stored with CG.
- **In**: the delay in frames from trigger until the graphic is taken. The default is 0, meaning no delay.
- **Duration**: the number of frames the graphic stays on air, used when **Graphics Out On** is set to *Timecode*. The default is 125 frames.
- **Graphics Out On**: when the graphic is taken off air. Nothing is selected by default, and an overlay saved without a choice is stored as *Timecode*, so it goes out after **Duration**. Choose one of:
    - **Timecode**: taken out after the configured Duration.
    - **Open End**: stays on air until something else takes it out.
    - **Manual**: taken out manually by the operator.
    - **Background End**: taken out when the background item ends. The **Out on** column of the table shows this mode as *Item end*.
    - **Story End**: taken out when the story ends.

Info: **Duration** only affects playout when **Graphics Out On** is *Timecode*. With any other out mode the value is stored but does not control when the graphic leaves air.

Copyright © 2026 Vizrt

Re-editing the Graphic Only

To change the graphic rather than its overlay settings, use **Open graphic** on the row. The Pilot Edge provider panel opens, if it is not already open, with the graphic loaded for editing. Make your changes in the Pilot Edge interface and send the graphic back, and the overlay is updated with the new graphic data.

13.3.6 Managing Overlays

Search

Use the search field in the toolbar to filter overlays by text. The search matches the overlay name (slug), description, handler name, graphics ID, MOS ID, and template type. Press **Ctrl+K**, or **Command+K** on macOS, to focus the search field.

Filter

Three filter dropdowns narrow the table. Each one offers only the values actually used by your existing overlays, so the choices reflect your own overlay set rather than every value the editor supports:

- **All Handlers**: filter by handler name.
- **All Out Modes**: filter by the Graphics Out On mode.
- **All Providers**: filter by where the overlay came from, either Pilot Edge or Custom.

The overlay count next to the filters shows how many overlays are listed, so you can see at a glance how much a filter has narrowed the table.

Sort

Click any sortable column header to sort by that column. The sort dropdown in the toolbar offers the same orderings by name:

- **Newest first** (default): overlays appear in the order they were added, newest at the top.
- **Name (A-Z)**: alphabetical order by slug.
- **Handler**: grouped by handler name, then alphabetical by slug.
- **Provider**: grouped by provider, then alphabetical by slug.

Delete

To delete an overlay, click the **Delete** button on the row, or select the row and press **Delete**. A **Delete overlay** dialog asks you to confirm, naming the overlay, for example *Are you sure you want to delete LOWERTHIRD?*

> ⚠ **Note**: *Deleting an overlay cannot be undone***.** The confirmation dialog is the only safeguard, so check the overlay name before confirming.
>
> - To keep a copy you can restore later, first export the overlay set to XML.

Copyright © 2026 Vizrt

13.3.7 NamedCG Actions

The **Actions** section of the overlay dialog controls what Mosart does to other on-air graphics when this overlay is taken. Choose one of three options:

Action	What it does
None (default)	No action when this overlay is taken. Other graphics are left alone.
Clear	Takes out all on-air graphics. Use it for overlays that need a clean screen before appearing.
Take out	Takes out stored on-air graphics matching a criterion, chosen below.

With **Take out** selected, choose which graphic to remove:

Option	Description
Last	Takes out the last graphic that was taken on air.
Last locator	Takes out the last locator graphic.
Last manual	Takes out the last manually triggered graphic.
Custom regex	Takes out all on-air graphics whose slug matches a regular expression. You can add multiple patterns.

Whatever you choose is shown in the **Action** column of the table, so you can see at a glance which overlays clear or take out other graphics.

13.3.8 NCS Content Placeholders

The **NCS Content** section of the overlay dialog adds placeholders that resolve from the newsroom system at runtime, so a single overlay can carry values from whichever story or item it is played in.

Placeholders are switches, grouped by scope. Turn on the ones you want the overlay to carry:

Scope	Placeholders
Story	Rundown ID, Story ID, Slug
Item	Rundown ID, Item ID, Slug, Type, Variant, Template

Each group also has a **Custom XPath** field with an **Add custom placeholder** button, for pulling a value the standard placeholders do not cover. Enter the XPath expression and add it to the group.

Copyright © 2026 Vizrt

13.3.9 XML Import and Export

Export

To export the overlays as an XML file:

1. Click the **Export XML** button in the sidebar, below the separator.
2. A file named `NamedOverlayGraphics.xml` is downloaded to your computer.

The exported file contains the overlays the table is currently showing, with their properties, MOS XML content and configured actions. Export writes only the listed rows, so clear the search field and the three filters first when you want the whole set. Use the file for backup, for migration between environments, or to import into another instance of the Named Overlays Editor.

Import

To import overlays from an XML file:

1. Click the **Import XML** button in the sidebar.
2. Select an XML file from your computer.
3. If the file contains overlays whose slugs already exist in the editor, a confirmation dialog lists the duplicates and offers **Skip Duplicates**, which keeps the existing overlays, or **Replace Duplicates**, which overwrites them with the imported versions. **Cancel** abandons the import.
4. A notification summarizes how many overlays were added, skipped, or replaced.

> **Info**: Import accepts both the editor's own export format and the legacy Mosart `NamedOverlayGraphics.xml` format. An overlay keeps the **Pilot Edge** provider when the file records it, which the editor's own export does. Everything else is imported as a **Custom** provider overlay.

13.3.10 Keyboard Shortcuts

Select **Keyboard Shortcuts** from the help menu in the toolbar to view this list in the editor.

Keys	Action
Ctrl+K, or Command+K on macOS	Focus search
N	New custom overlay
Up arrow / Down arrow	Navigate list
Enter	Open properties
Delete, or Backspace on Windows	Delete selected overlay
Escape	Close panel or clear selection

Copyright © 2026 Vizrt

Info: **N** is a plain key press, without **Ctrl** or Command, because browsers reserve **Ctrl+N** for a new browser window.
Apart from **Ctrl+K**, or **Command+K** on macOS, which always focuses the search field, shortcuts are ignored while a text field, dropdown, or dialog has focus, so they cannot interfere with typing. The shortcuts that create, delete, or modify overlays do nothing for users with only a *Viewer* role.

Copyright © 2026 Vizrt

13.4 Setup and Administration NOE

This page describes how to install, configure, and administer the Named Overlays Editor in a Viz Mosart deployment.

- Prerequisites
- Accessing the Editor
- Authentication and Roles
- Configuring the Viz Pilot Edge Plugin
- Main and Backup Servers
- Troubleshooting

13.4.1 Prerequisites

Requirement	Details
Mosart Web Apps	The Named Overlays Editor is served by the Web Apps server and is included in the Mosart Web Apps installation package. The server must be installed and running.
Viz Mosart automation system	A running Mosart server is required. The editor reads and writes the named overlays on the Mosart server, and the toolbar shows a beacon for the main server, plus one for the backup server in a redundancy setup, with each server's own state.
Viz Pilot Edge (optional)	Required only if you want to import graphics from a Viz Pilot Data Server. Pilot Edge is a web plugin hosted by the Pilot Data Server.
Web browser	A modern web browser. The editor is a single-page application that runs entirely in the browser.

13.4.2 Accessing the Editor

Once the Web Apps server is running, the Named Overlays Editor is served at: *https://<server-address>/namedoverlayseditor/*

Replace `<server-address>` with the host name or IP address of the machine running the Web Apps server. The editor can also be opened from **Home** in the Mosart Web Apps Configuration Tool.

On an installation with more than one studio the editor works on one studio at a time. **Home** launches it with the studio selected there, appended to the address as `?studioId=<id>`, and the toolbar shows the active studio with a dropdown to switch. Opened without that parameter, the editor falls back to the first configured studio. Named overlays live on each studio's own Mosart server, so the overlay set changes with the studio.

If authentication is enabled, you are redirected to a sign-in page before the editor opens. See *Authentication and roles* below.

Copyright © 2026 Vizrt

13.4.3 Authentication and Roles

The editor recognises two roles. You assign them in your identity provider, not in the Configuration Tool, and the editor reads them from the `roles` claim in the user's token.

Role name	What the user can do
`mosart-admin`	Everything: view overlays, create, edit, delete, import and export them, update a graphic from the provider panel, and sync overlays to the backup server.
`mosart-viewer`	View the overlay list, and search, filter, sort and export it. No create, edit, delete, import, or sync, and the overlay dialog does not open: the row buttons and the **Pilot Edge**, **Custom** and **Import XML** buttons are not shown.

Assign the roles before you enable authentication

A signed-in user who holds neither role does not get a read-only view. The editor shows an unauthorized-access screen and nothing else, because it fails closed. An identity provider with no Mosart roles configured therefore locks out everybody, which reads as a fault in the editor rather than as missing configuration. Assign a role to every user who needs the editor before you turn authentication on.

The roles must reach both tokens

Two layers check the user, and they read two different tokens:

Layer	Reads	Decides
Mosart Web Apps server	the ID token	whether the editor opens at all, and whether the editing controls are shown
Mosart server	the access token	whether the REST API accepts the request

This is the most common misconfiguration. If the roles are present in only the ID token, the editor looks completely correct and every save fails.

The claim must be flat, top-level, and named exactly `roles`. A nested claim, such as Keycloak's default `realm_access.roles`, is not read: the Web Apps server matches the claim name exactly.

Confirm that your provider emits the claim in both tokens. In Keycloak this means adding a protocol mapper for the roles and enabling it for the ID token and the access token separately. In Microsoft Entra ID, app roles that you have defined and assigned are emitted in a `roles` claim with no further claim mapping.

Copyright © 2026 Vizrt

Where roles are enforced

Hiding the editing controls in the browser is not the whole protection. The Mosart server checks the role on every named overlays REST call, so a viewer cannot create, change, or delete an overlay by calling the API directly.

> **Info**: Enforcement on the Mosart server requires Viz Mosart 5.16 or later with Mosart Web Apps 3.2 or later, and OIDC authentication enabled on the Mosart server itself, under *Remote Controller Service* > *Security* in the Mosart settings.

With authentication disabled, every user is treated as an administrator, no request is refused, and the roles have no effect.

Session Timeout

Where authentication is enabled, the Named Overlays Editor signs a user out after a period without activity. This started in Mosart Web Apps 3.3. Before that, an editor left open held its session for as long as the page was open, because its own background token refresh kept the session alive. That background traffic no longer counts as activity.

> **Note**: Expect support questions from sites that leave the editor open on a spare screen. Nothing is lost, but a user returning to it now finds a sign-in dialog where none appeared before.

The controls are on the **Server Configuration** page of the Configuration Tool, inside the **Authentication (OIDC)** block, under **Allow session timeout**:

- The **Named Overlays Editor** switch is on by default, reading *Renews while in use; can time out when idle*. Switch it off to read *Stays signed in while open*, which restores the earlier behaviour.
- **Session timeout** sets how long an unused session survives, in minutes, from 5 to 10080, defaulting to 30. It applies only to the apps whose switch is on.

Activity means typing, clicking, scrolling and moving the pointer in the editor. The embedded Viz Pilot Edge panel is a separate frame, so its events are invisible to the editor: having the panel in focus stands in for activity for five minutes after the last action in the editor and no longer, while a graphic sent back from the plugin counts as activity in full. A long spell spent only inside the panel can therefore still reach the countdown. Ninety seconds before the session ends, a dialog headed **You will be logged out soon** offers **Continue session** or **Log out now**; if that is missed, the editor shows **Session timeout!** with a **Login** button. See Authentication.

Messages you may see

Message	What it means
The Mosart Server refused this operation for your user.	You are signed in, but your user does not hold the role this operation needs. Viewing needs `mosart-viewer` or `mosart-admin`, changing anything needs `mosart-admin`. Ask an administrator to assign the role in the identity provider. This is not a certificate or key problem.

Copyright © 2026 Vizrt

Message	What it means
The Mosart Server rejected the request as unauthenticated.	No valid credential reached the Mosart server. An expired session now shows the **Session timeout!** dialog in its own right, so if you are seeing this message instead, the sign-in is present but the Mosart server would not accept it. Only when authentication is disabled does this point at the REST API key, which must match the key in the Configuration Tool. While OIDC authentication is on, the Mosart server ignores the REST API key entirely, so changing that key does not fix this.
No active Mosart Server.	Not a permissions problem. Automation must be running and on air on the main or the backup server before overlays can be loaded or saved.

13.4.4 Configuring the Viz Pilot Edge Plugin

The Pilot Edge plugin panel is configured in the Configuration Tool, on the **Named Overlays Editor** page of **App Configuration**:

1. In the Configuration Tool, open **App Configuration** and select **Named Overlays Editor**.
2. Check that **VIZ PILOT GRAPHICS AND ELEMENTS** is selected. It is selected when the page opens: the tool fills it in from Showmaker's Pilot Edge configuration when one exists, and from the defaults below otherwise. Clear it if this editor should have no plugin panel.
3. Provide values for:
   - **Application URL**: where Viz Pilot Edge is running. Default: *http://localhost:8177/app/pilotedge/pilotedge.html?mosid=pilot&sendbutton*. Enter the host of your Viz Pilot Edge installation if it is not on the same machine.
   - **MOS Plugin ID**: the MOS ID stamped on overlays created from this plugin. Default: *PILOT*.
4. Click **Save**.

Tip: The settings are stored in the `NamedOverlaysEditor` section of the Web Apps server configuration, under `MosPlugins`, with the plugin type `pilot`. Editing the file by hand is not required: the Configuration Tool writes it for you.

Info: The plugin panel is optional. With no plugin configured, the editor still works for creating custom overlays and for XML import and export.

13.4.5 Main and Backup Servers

The toolbar carries a beacon and a label for the main Mosart server, and a second pair for the backup server in a redundancy setup. Each label is that server's own name as Viz Mosart reports it, falling back to **Main** and **Backup** when no name is available, and each beacon shows that server's own state, so the green one is the server reported as Active. The editor is routed to whichever server is active; it does not choose between them.

In a redundancy setup, named overlays live on both servers, and the two can drift apart if overlays are edited while one of them is unavailable. To align them, use **Sync to backup server** in the sidebar. It makes the backup server's

Copyright © 2026 Vizrt

overlays identical to the main server's: every overlay on the main server is copied across, replacing any overlay with the same slug, and overlays that exist only on the backup are removed. A confirmation dialog appears first, and a notification reports how many overlays were copied and removed.

> ⚠ **Note**: The **Sync to backup server** button is only shown to administrators, and only when the editor detects that a backup server is configured. In a single-server installation it does not appear at all.

The sync always reads from the main server and writes to the backup server explicitly, so it is safe to run during a failover: it will not copy the backup onto itself. Because it removes overlays that exist only on the backup, make sure the main server holds the set you want before running it.

Where overlays are stored

Overlays are stored on the Mosart server in `NamedOverlayGraphics.xml`, not in the browser and not on the Web Apps host. Any editor instance pointed at the same Mosart server sees the same overlay set, and the overlays are available to Mosart automation as soon as they are created. To keep a copy outside Mosart, use **Export XML** in the editor.

For background on named overlays and how Viz Mosart uses them, see *Named Overlay Graphics* in the Viz Mosart Administrator Guide. The editor links to it directly from **Help** in its help menu.

13.4.6 Troubleshooting

Symptom	What to check
The toolbar does not show a connected Mosart server	Check that the Mosart server is running and reachable, and that the server connection is correctly configured in the Configuration Tool under *Server Configuration*.
The overlay list stays empty on a server that has overlays	The editor cannot reach the Mosart server. Check the connection indicator in the toolbar and the server configuration.
The provider panel opens but stays empty	The plugin URL is not reachable from the browser. Open the URL directly in the same browser to confirm, and check that the Pilot Data Server is available.
Thumbnails are missing for Pilot Edge overlays	Thumbnails are resolved from the Pilot Data Server. Check that it is reachable from the browser.
Editing controls are missing	The signed-in user holds `mosart-viewer` rather than `mosart-admin`. See *Authentication and roles* above.

Copyright © 2026 Vizrt

Symptom	What to check
The editor shows an unauthorized-access screen instead of opening	The signed-in user holds neither `mosart-admin` nor `mosart-viewer`. Check that the role is assigned in the identity provider, and that it is spelled exactly as above: the earlier `namedoverlays-` names are no longer recognised.
The editor opens and looks correct, but every save is refused	The roles most likely reach the ID token but not the access token. The Mosart server reads the access token. See *The roles must reach both tokens* above.
Reads work but writes are refused for an administrator	Confirm the role name in the token is `mosart-admin`, and that the Mosart server was restarted after OIDC authentication was enabled. Every setting under *Remote Controller Service > Security* requires a restart of the Remote Control Service.
The editor signs a user out while nobody is using it	Expected since Mosart Web Apps 3.3: the editor times out when idle. To restore the earlier behaviour, switch **Named Overlays Editor** off under **Allow session timeout** on the *Server Configuration* page of the Configuration Tool. See *Session Timeout* above.
The **Sync to backup server** button is missing	It appears only for administrators, and only when a backup server is detected. Check the redundancy configuration for the Mosart server.

For general connectivity issues, see section Troubleshooting.

Copyright © 2026 Vizrt

14 Troubleshooting

- Where to start
- Web Apps Configuration Tool
- Sign-in and permission issues
- Slow loading or a client limit in the live apps
- Issues with Mosart Web Server Connecting to the Mosart Server
- Browser Logs

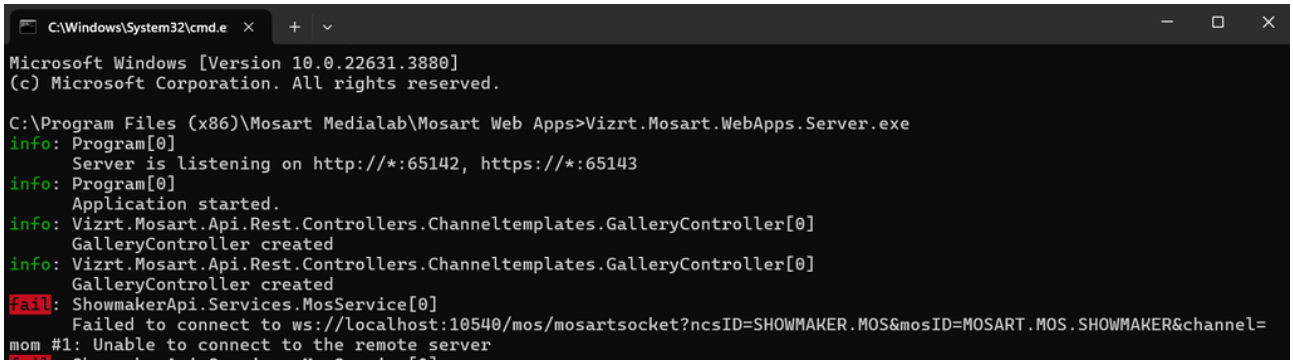
14.1 Where to start

Symptom	Look at
The Configuration Tool reports a service as offline, or a page does not load at all	Web Apps Configuration Tool, below: the proxy, the web applications service, or the external proxy
Apps open, but no rundown data or timers arrive	Unable to reach the Viz Mosart Server, and Issues with Mosart Web Server Connecting to the Mosart Server
Content loads slowly, or only the first 5 to 6 clients work	Slow loading or a client limit in the live apps
Sign-in problems, an unauthorized-access screen, an unexpected logout, or a save that does not reach the server	Sign-in and permission issues
503 or 504 from a studio route	Not a fault in the Web Apps: no Viz Mosart server in that studio's pair is currently on air

14.2 Web Apps Configuration Tool

The *Mosart Web Applications Configuration Tool* provides you with error or warning messages that assist with common issues.
These messages appear in the status belt at the top of the tool, and most offer **Web server log** and **Proxy server log**. A user without the right to change settings sees how to get that right instead. In a browser, each link shows the most recent lines of that log in a dialog. In the Configuration Tool window on the server, it opens the log folder instead.

Copyright © 2026 Vizrt



To troubleshoot issues, check the following locations for relevant logs:

- **Web server logs**: By default, these are written to `C:\MMLogs\WebServer`. The location is configurable in the web server configuration file: `C:\ProgramData\Mosart Medialab\Mosart Web Apps\serverSettings.json.` Look for the `NLogConfig` setting.
- **Proxy server logs**: By default, these are stored at `C:\MMLogs\Mosart Web Proxy`.
- **Windows Event Log**: Use the Event Viewer to check for system or application-related errors.
- **Browser logs**: To view logs in your browser, press **Ctrl+Shift+I** (this toggles the Developer Tools panel). Do this in a browser with the **Mosart Web Applications Configuration Tool** or a Mosart web application open. The Configuration Tool window on the server has no developer tools, so open the tool in a browser for this.

14.2.1 The Configuration Tool is refused, or shows only Home

Symptom	Cause
The Config Tool is available on the Mosart server only.	The tool was opened from another computer while **Serve Config Tool on the web** is off. Open it on the Web Apps server, or turn the setting on in Server Configuration.
The Config Tool answers only on this server's own host names	The tool was opened on the server under another name, such as an alias, while sign-in is off. Open it on `localhost`, the machine name or address, or a host name set in Server Configuration.
Only **Home** is available, and the other pages are dimmed	You do not have the right to change settings. With sign-in on, that takes the `configtool-admin` role. With sign-in off, settings are changed only in the Configuration Tool window on the server, run by a Windows administrator. See Configuration Tool.

Copyright © 2026 Vizrt

14.2.2 The Mosart Web Proxy Service is not running or is not functioning properly

Review the settings in

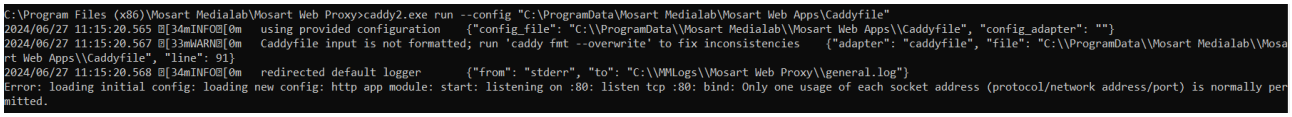
- The Server Configuration page.
  Proxy server logs are located by default at C:\MMLogs\Mosart Web Proxy.
- The browser logs.
  You can investigate your browser logs by typing **CTRL+Shift+I** (this is a toggle) in the browser where the **Mosart Web Applications Configuration Tool** or a Mosart web application is open.

Start the Mosart Proxy Service in console

If the Mosart Web Proxy service fails to start and there is little information in either the proxy logs or the browser, you can start Caddy in the console to troubleshoot the problem using the following command. If this is running as a Windows service, stop it first.

```
C:\Program Files (x86)\Mosart Medialab\Mosart Web Proxy>caddy2.exe run --config "C:
\ProgramData\Mosart Medialab\Mosart Web Apps\Caddyfile"
```

Typical situations are where Caddy fails to start because the port (default **55142** for HTTP and **55143** for HTTPS) is already in use by other applications or the certificate and/or the key files configured are not valid:



14.2.3 The Mosart Web Applications Service is not running or is not functioning properly

A common *offline* error occurs when the Windows service fails to start because its port is already in use. The default port is **65142** for HTTP and **65143** for HTTPS, and it can only be changed in the configuration file at *C:\ProgramData\Mosart Medialab\Mosart Web Apps\serverSettings.json*.

> **Note**: **Save** in the Mosart Web Applications Configuration Tool writes back the settings as the tool loaded them, so a value changed by hand in the Viz Mosart web server configuration file (C:\ProgramData\Mosart Medialab\Mosart Web Apps\serverSettings.json) while the tool was open can be overwritten. Reload the tool after editing the file. Each save first keeps a copy of the file as it was, beside it, named with the date and time, for example `serverSettings-20260717-153045123.json`. The 20 most recent copies are kept.

Copyright © 2026 Vizrt

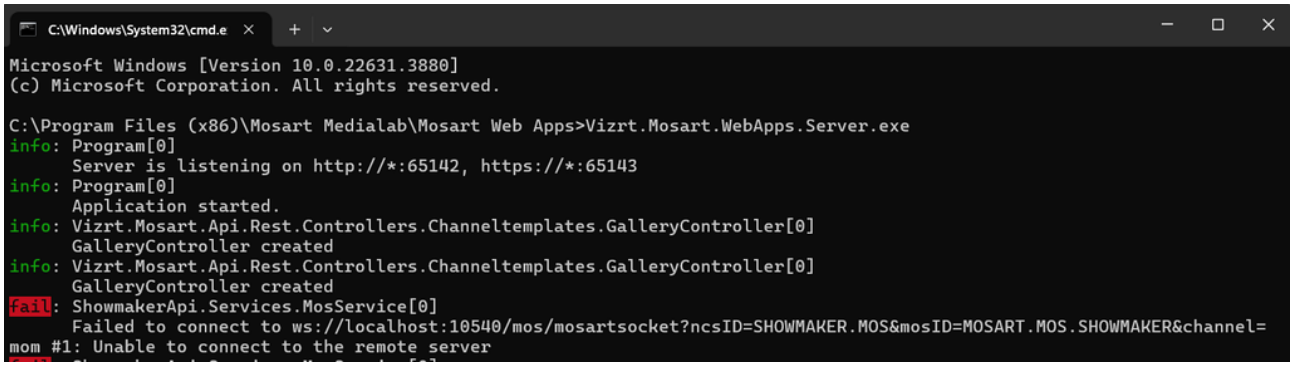
Review the settings in

- The Server Configuration page.
- The web server log files, from **Web server log** in the status belt at the top of the *Mosart Web Applications Configuration Tool*.
- Web server logs written on disk (default location "*C:/MMLogs/WebServer*" configurable in the web server configuration file, *C:\ProgramData\Mosart Medialab\Mosart Web Apps\serverSettings.json* - search for setting *NLogConfig*).
- The browser logs. You can investigate the browser logs by typing **CTRL+Shift+I** (this is a toggle) in the browser where the *Mosart Web Applications Configuration Tool* or a Mosart web application was opened, for more details.

You can also try restarting the service with **Restart service** in the status belt at the top of the Configuration Tool. In a browser this works only while the Web Apps server still answers. The Configuration Tool window on the server can also start a service that has stopped, and when the service will not start, it opens the tool in settings-only mode so that you can correct the setting. See Configuration Tool.

Start the Mosart Web Applications Service in console

If the Mosart Web Applications service fails to start and there is little information in either the web server logs or the browser, you can start the service in the console to troubleshoot the problem. If this is running as a Windows service, stop it first.



Typical situations are where the web server fails to start because the port (default **65142** for HTTP and **65143** for HTTPS) is already in use by other applications or the certificate and/or the key files configured for HTTPS are not valid.

14.2.4 The External Proxy Service is unreachable or not functioning properly

Review the settings in

- The Server Configuration page.
- External proxy configurations.
- The web server logs, from **Web server log** in the status belt at the top of the *Mosart Web Applications Configuration Tool*.

Copyright © 2026 Vizrt

- The browser logs. You can investigate the browser logs by typing **CTRL+Shift+I** (this is a toggle) in the browser where the Mosart Web Applications Configuration Tool or a Mosart web application was opened.

14.2.5 Unable to reach the Viz Mosart Server (main or backup)

Some of the Mosart Web Applications might not function as expected. Ensure that the *Mosart Remote Control Service* (RCS) located on the Viz Mosart server is running. Review the settings in the Server Configuration page, web server log file, from **Web server log** in the status belt of the Mosart Web Applications Configuration Tool, or the browser logs for more details. You can investigate the browser logs by typing **CTRL+Shift+I** (this is a toggle) in the browser where the Mosart Web Applications Configuration Tool or a Mosart web application was opened.

Check the status of the Viz Mosart server

In a browser, paste the URL to the Viz Mosart server *status* endpoint:

HTTP
http://<mosart-server-host>:55167/status for HTTP

HTTPS
https://<mosart-server-host>:55168/status for HTTPS

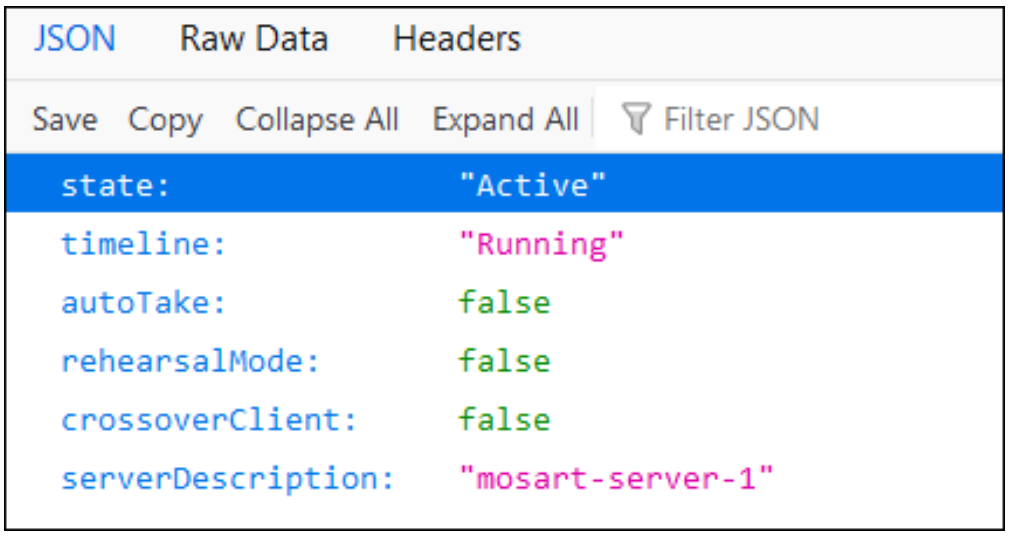
Using a Web proxy

http://<webproxy-url>:55142/mosart/main/status

> **Note**: In the first two examples above, replace *<mosart-server-host>* with the hostname, IP address or the FQDN of your Viz Mosart server.

> **Info**: `/status` needs no credential, even when the Mosart server runs with authentication enabled, so it stays usable as a health check. The versioned alias `/api/v1/status` does require one.

You should get a result similar with the following (*serverDescription* is not available in older Mosart versions):



Copyright © 2026 Vizrt

If no response is received:

- Ensure that the Viz Mosart server version is compatible with the Viz Mosart Web Applications. Refer to the Release Notes for System Requirements.
  or
- Start Mosart Remote Control Service (RCS) as console to investigate further the issue. If the service is running as Windows service, stop it first.
  Typical situations are where RCS fails to start because the port (default **55167** for HTTP and **55168** for HTTPS) is already in use by other applications or the certificate and/or the key files configured for HTTPS are not valid:

Viz Mosart Server version might be incompatible with this version of Viz Mosart Web Applications

- Please review the corresponding version Release Notes for product interoperability.

If HTTPS is used, this must be enabled on both Viz Mosart Server and Viz Mosart Web Applications

You may get these warning messages because no updates could be retrieved from Viz Mosart server.

- Ensure that the Viz Mosart server version is compatible with the Viz Mosart Web Applications. Refer to the latest Release Notes for System Requirements.
- Ensure that HTTPS is used and enabled on both Viz Mosart server and the Viz Mosart Web Applications. Check the settings in the Server Configuration page, Viz Mosart web server log file or the browser logs (**CTRL+Shift+I** in a browser with the Configuration Tool open) for more details.
- Check the status to the Viz Mosart server as described above.

If the server answers on its status endpoint but requests through the web apps fail with 500 and authentication is enabled, the cause is usually on the Mosart server itself: it cannot reach the identity provider. See the *Viz Mosart Administrator Guide*, chapter *Security in Viz Mosart Ecosystem*.

14.3 Sign-in and permission issues

These apply when authentication is enabled. The web apps read the user's roles from the token your identity provider issues; roles are never assigned in Viz Mosart. For the full setup and the identity provider requirements, see the *Viz Mosart Administrator Guide*, chapter *Security in Viz Mosart Ecosystem*, and the Authentication page.

Symptom	Cause
A signed-in user gets an unauthorized-access screen	That user holds no role for the app. Assign one in the identity provider. Note that a user with no role does not get a read-only view.

Copyright © 2026 Vizrt

Symptom	Cause
Everybody is locked out right after enabling authentication	No roles assigned yet, or the role names do not match. The Named Overlays Editor roles are `mosart-admin` and `mosart-viewer`; earlier documentation used `namedoverlays-admin` and `namedoverlays-viewer`, which no current version reads.
An app looks correct, with all its controls, but every save fails	The roles reached the ID token only. They must be present in **both** the ID token and the access token: the Web Apps server reads the ID token to decide what the app shows, the Mosart server reads the access token to accept or refuse the call.
An unattended screen, such as a wall display, shows a login page some hours after it was started	Either the `offline_access` scope is missing in the **Scopes** field, so the token the app uses to reach the Viz Mosart server cannot be refreshed, or that app has been switched on in **Allow session timeout** in Server Configuration, so it now expires when nobody is at it. For an unattended screen, set it back to **Stays signed in while open**.
Session timeout! reading *You have been logged out due to inactivity.*	Expected for an app that is switched on in **Allow session timeout**, by default Showmaker, the Template Editor and the Named Overlays Editor, when nobody has been active for the length of **Session timeout**, 30 minutes by default. Raise the window, or set that app to **Stays signed in while open**.
Session timeout! reading *The connection to the server was lost for too long, so you have been logged out.*	An app set to **Stays signed in while open** could not reach the Mosart Web Apps server for longer than **Session timeout**. This is a connection fault rather than a timeout setting: check the proxy and the web applications service, above.
A wall panel shows **Session timeout!** reading *Your session has ended. Log in to continue.*	The page was loaded onto a session that had already ended: the machine had been switched off or rebooted, or the page was reloaded. Nobody was signed in on that page for anything to expire.
You will be logged out soon appears while somebody is working	From 3.3 the window follows the user's own activity, so this should not happen at a keyboard. Check whether the user works mainly inside an embedded panel from another product, such as Viz Pilot Edge: the host page cannot see that panel's events, so focus alone stands in for activity for five minutes at a time.
A laptop reopened after a long sleep goes straight to **Session timeout!** with no warning first	Expected. The countdown comes from the server, not the browser clock, so a machine that was asleep through the warning window finds the session already gone.

Copyright © 2026 Vizrt

Symptom	Cause
The NRCS Plugin stays blank inside ENPS, with no certificate warning	Either the Content Security Policy on the identity provider does not list the ENPS host under `frame-ancestors`, or the certificate is not trusted on that machine. An embedded plugin often shows no warning at all, just a blank page.
Mosart Server settings under Help & Resources is dimmed	The Remote Control Service on that Mosart server is not reachable, or the settings editor is not enabled on it, or OIDC authentication is enabled on the Mosart server and the editor has no browser sign-in. The row's own tooltip says which it is. It is also dimmed for a user without the right to change the configuration. See Configuration Tool.
Settings for Manus Administrator shows a warning instead of offering to push the settings, or a push ends with **Failed to apply new settings.**	The Configuration Tool could not read the settings from the main Viz Mosart server, or the push failed. The message names the reason. For each message and what to do, see App Configuration, under *Showmaker > Connection settings*.
An integration that used to work now returns 401	Expected if it authenticates with the Mosart REST API key: enabling OIDC on the Mosart server switches the key off.

14.3.1 A save did not reach the server

Showmaker talks to the server over a real-time connection. Before 3.3 that traffic did not renew an idle session, so once the session had gone the saves were refused with nothing on screen to say so, and a journalist could lose a script on reload. From 3.3 the failure is visible and the work is kept.

- While the connection is down, Showmaker shows a banner at the top of the rundown: **Disconnected from the server. Changes are not being saved.** It appears only once a connection has succeeded, so it does not flash during start-up.
- A save the server refuses raises an error toast naming what was lost, **Could not save the script**, **Could not save the story**, **Could not save the item** and so on, followed by the server's own reason.
- Showmaker keeps retrying the connection, and asks the server about the session as it does. A session that has really ended therefore produces the **Session timeout!** dialog rather than an app that goes on looking signed in.
- Unsaved script text is committed automatically in the last seconds before the session ends, so in the ordinary case there is nothing to recover.
- If no save could run at all, because the machine was asleep or offline for the whole countdown, the text is kept in that browser tab and put back when you sign in again. If the same line was also changed by somebody else while you were signed out, a warning toast says so: *Recovered your unsaved text. This line was also changed while you were signed out, so check it before saving.*

Copyright © 2026 Vizrt

Info: If a user reports lost work on a version before 3.3, this is the mechanism to look for. It was reported as support case 00227212.

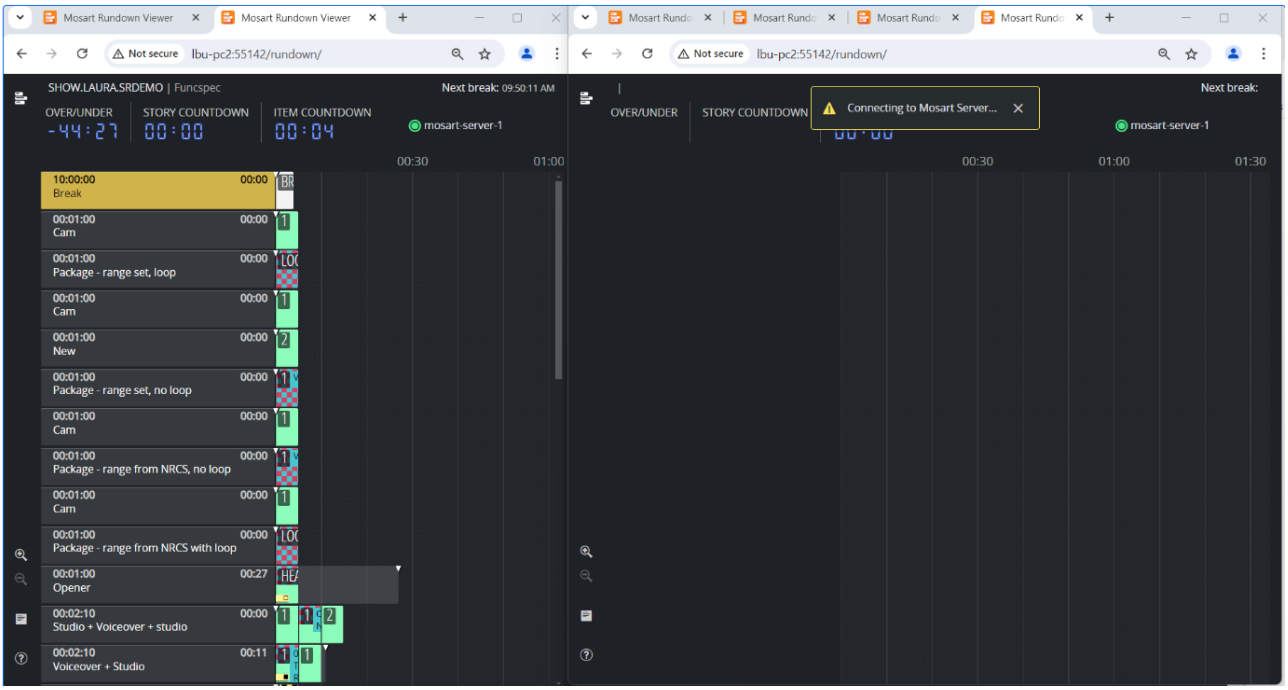
14.4 Slow loading or a client limit in the live apps

If you experience the following issues with the live apps (Rundown Viewer, Smart Panel, Prompter):

- The content loads slowly or only partially.
- You cannot open more than 5-6 web sessions, meaning that opening the web application in multiple browser tabs or instances fails after 5-6 clients.

These problems may indicate that HTTPS is not properly configured across all relevant components.

In the screenshot below, the 6th Rundown Viewer client displays an error connecting to the Mosart server, while the other clients are functioning correctly:



Troubleshooting Steps:

1. **Inspect Browser Logs:**
   - Open the browser's Developer Tools and go to the **Network** tab.
   - Inspect some of the URLs. For example, you might see a request URL to the Mosart Server like `http://proxy-server:55142/mosart/mosarthub/negotiate?name=MosartWebUI&negotiateVersion=1`.
   - A "307 Temporary Redirect" status code indicates that the resource has been temporarily moved to a different URL. This suggests that the proxy server (Caddy) has redirected the client to use HTTPS instead of the original HTTP request.

Copyright © 2026 Vizrt

2. **Understand the issue:**
    - In this example, the client (the Rundown Viewer in the web browser) initially made a request over HTTP to the Mosart server, but the proxy server redirected it to HTTPS.
    - This indicates that HTTPS is configured for the Mosart server but not for the Mosart web server and the proxy server.
    - By default, the Mosart Web Proxy is configured to enforce HTTPS by redirecting HTTP requests to HTTPS before they reach the Mosart server, if the Mosart server is set up with HTTPS.
3. **Fix the problem:**
    - To resolve this issue, enable HTTPS for the Mosart web server and the proxy server (internal or external) using the Mosart Web Apps Configuration Tool.

14.5 Issues with Mosart Web Server Connecting to the Mosart Server

14.5.1 Mosart Remote Control Service (RCS)

Issues with Mosart web server connecting to the Mosart Remote Control Service (RCS), may be due to the RCS not automatically detecting the fully qualified domain name (FQDN) of the machine where it is installed.

For enhanced security and connection stability between the Mosart web server and the Mosart Remote Control Service, the setting **SignalR allowed origins** lets you specify an exclusive list of origins that may access the SignalR service used for communication with the RCS.

To configure it, open the Mosart settings on the Viz Mosart server, from Manus Administrator (type *settings* in the console), AV Automation or Media Administrator, and go to **Remote Controller Service > Security**:

1. Set **SignalR allowed origins** to a semicolon-separated list of origins, including the fully qualified domain name of your Mosart server, for example `https://app.example.com;https://admin.example.com`. Left empty, the auto-detected host names are used.
2. Restart the Mosart Remote Control Service. The setting is marked "Requires restart".

Note: For Mosart applications that rely on SignalR for real-time communication, such as the Rundown Viewer, Smart Panel and Prompter, defining the allowed origins is essential when a failure to recognize an FQDN prevents communication with the RCS.

Info: In earlier versions this value was set as `SignalRAllowedOrigins` by hand in the RCS configuration file *RemoteDispatcherServiceConfig.xml*. Use the setting above instead; a value edited directly in the installation folder can be replaced on upgrade.

14.5.2 Firewall Ports

- Ensure the default ports have access.

Copyright © 2026 Vizrt

14.6 Browser Logs

14.6.1 To investigate browser logs

In failure situations, you can use your web browser to access the Mosart Web Apps directly and further investigate the browser logs.

1. From **Home** in the Configuration Tool, click a web app to open it in the default browser, or paste the web app link directly into your preferred browser:

Application	Address
NRCS Plugin	*http(s)://<your-hostname>:<web-proxy-port>/plugin/?nrcs=ENPS&mosid=mosart*
Rundown Viewer	*http(s)://<your-hostname>:<web-proxy-port>/rundown*
Smart Panel	*http(s)://<your-hostname>:<web-proxy-port>/smartpanel*
Prompter	*http(s)://<your-hostname>:<web-proxy-port>/prompter*
Showmaker	*http(s)://<your-hostname>:<web-proxy-port>/showmaker/*
Named Overlays Editor	*http(s)://<your-hostname>:<web-proxy-port>/namedoverlayseditor*
Template Editor	*http(s)://<your-hostname>:<web-proxy-port>/templateeditor*

   In a multi-studio installation, add `?studioId=<studio>` to open a studio-scoped app against that studio.
2. Type **CTRL+Shift+I** (this is a toggle) in the browser and inspect the **Console** tab or the **Network** tab.

Copyright © 2026 Vizrt

15 Appendix

This appendix collects reference material that sits outside the main setup and operating guides.

- Vizrt SSO (Keycloak): connecting the Mosart Web Applications to a Vizrt SSO, for example when running alongside a Viz One that is authenticated by SSO. Covers importing the realm configuration, the client roles and groups it creates, and the OIDC settings to enter in the Configuration Tool.
- Elgato Stream Deck Plugin: installing the Stream Deck plugin, pointing it at the Mosart Web Apps host, and assigning Viz Mosart commands to its keys.

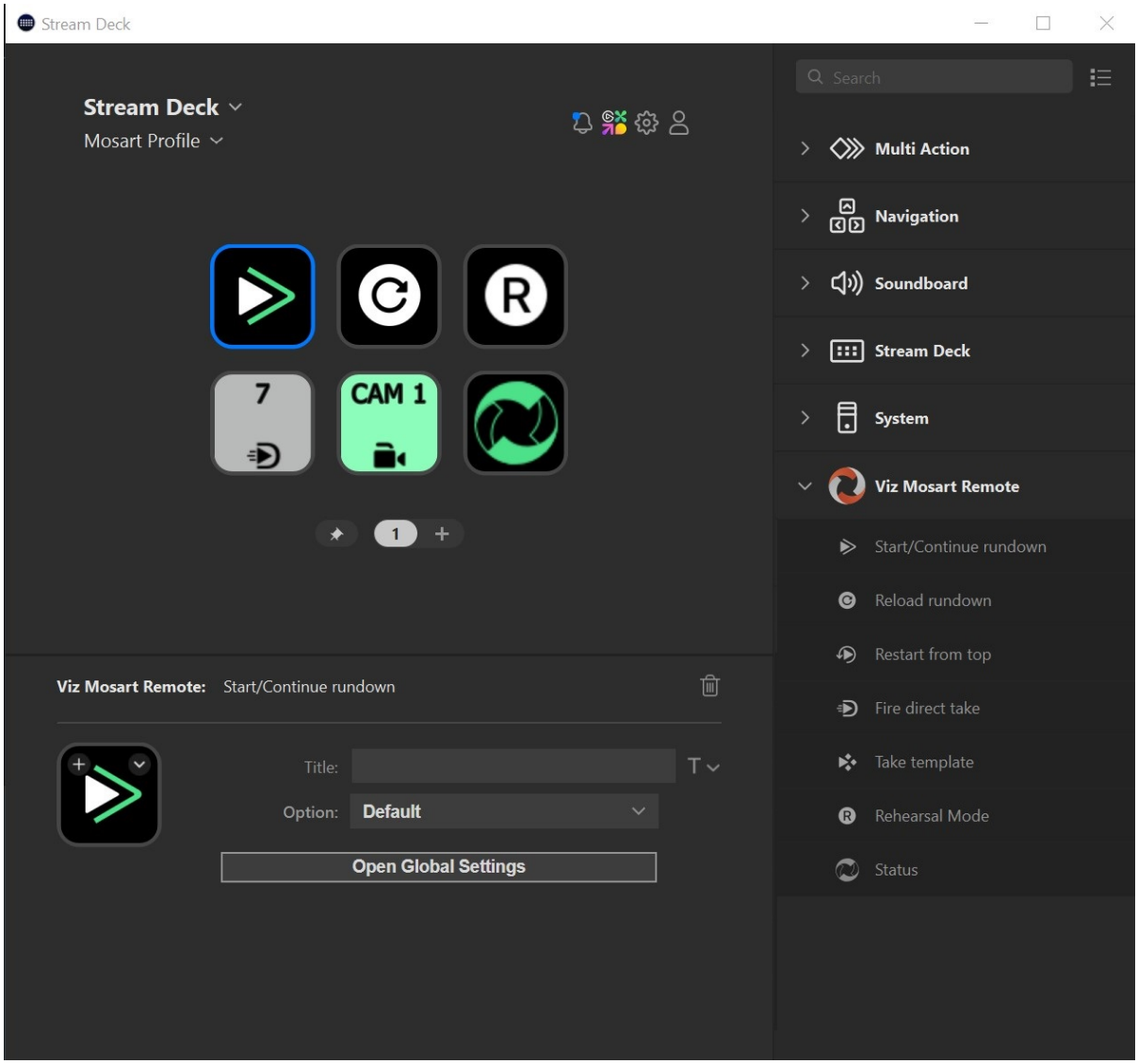
For the Web Apps half of authentication, including the roles each application enforces, see Authentication.

Copyright © 2026 Vizrt

15.1 Elgato Stream Deck Plugin

The Elgato Stream Deck plugin for Viz Mosart allows you to send Viz Mosart rundown actions including *Start/Continue*, *Skip Next*, *Unskip Next*, *Take a template*. You can also fire a *Direct take* or a *Control* command.

The plugin communicates directly with the Viz Mosart server and all actions are triggered without needing a Mosart GUI open, active or in focus.



- Installation
- Configuration
- Actions
- Status

Copyright © 2026 Vizrt

15.1.1 Installation

Software Download

To download Stream Deck for Windows

1. From Elgato's site, https://www.elgato.com/us/en/s/downloads
   select *Stream Deck* for *Windows* software and download.
   The Stream Deck utility starts up automatically after install.

> **Info**: The software can be installed on a different machine than the Viz Mosart Server or Mosart GUI client.

2. If not already installed, download and install the latest version of Viz Mosart Web Apps from https://download.vizrt.com/ at
   /products/VizMosart/Latest Version/WebApplications/
   and perform the one-time Server Configuration setup.

> **Note**: Viz Mosart Web Apps has to be installed if you want to use the Stream Deck plugin with Mosart

3. From the Web Apps installation folder, locate the plugin file *com.vizrt.mosart-remote.streamDeckPlugin*. Rather than hunting for it, open the Configuration Tool and select **Stream Deck Plugin** under **Help & Resources**: selecting it, or its **Download plugin** button, downloads the file. On an installed system the file sits in the *StreamDeckPlugin* folder of the Mosart Web Apps server's installation folder.
4. Double click the file *com.vizrt.mosart-remote.streamDeckPlugin* for installation on the *same PC where you just installed the Stream Deck* for Windows software.
5. Accept when the Stream Deck application requests installation confirmation.
6. After installation, the **Viz Mosart** section displays with newly available actions.

15.1.2 Configuration

Configuring the Mosart Host

To configure the Viz Mosart host

1. Drag the **Start/Continue rundown** to any free Stream Deck key.
2. Click **Open Global Settings** to configure the plugin.
3. In field **Mosart Url**, enter the URL of the hostname of where your Mosart Web Applications are installed (failover *is* supported).
   This Mosart Web applications proxy web server must be prefixed with *http://* and suffixed with port number *:55142*, or prefixed with *https://* and suffixed with port number *:55143*. See the screenshot for an example.

> **Info**: See the Viz Mosart Web Applications Config Tool User Guide (section *Server Configuration*) for more information about the *Mosart Web applications proxy web server*.

5. In field **X-API-key** enter the key.

Copyright © 2026 Vizrt

Info: See the Appendix topic *Mosart Remote Control REST API* in the *Viz Mosart Administrator Guide* for more information about *REST API endpoint* and *X-API-key*.
For redundancy setup, configure the *same* API-key on both main and backup.

6. Click **Check Server** to test the connection. The button turns green and reads *Success* when the check succeeds, and turns red and reads *Unable to find Mosart server* when it fails.
7. Under **Other settings**, **Show Template and Direct take icons by default** sets the icon default for every Take Template and Fire Direct Take key. It is on unless you clear it, and a key's own **Show icon** option overrides it.
8. Click **Save and Close**. (The settings are set once-only and are saved for all operations).

15.1.3 Actions

- Start/Continue Rundown
- Reload Rundown
- Restart from Top
- Fire Direct Take
- Take Template
- Take Overlay Graphics
- Rehearsal Mode
- Skip Next
- Unskip Next
- Control Command

Start/Continue Rundown

Start or Continue a rundown. This action is the equivalent of a Mosart GUI timeline key, S*tart Continue*.

Options

- **Default**: The next template's default transition.
- **Mix or Wipe**: Presents the *rate* entry field.
- **Effect**: The number of the effect.

Reload Rundown

This action triggers a rundown reload event. The timeline is stopped and the current rundown reloaded from the NRCS. No additional parameters.

Restart from Top

Restart the rundown from the first story. No additional parameters.

Fire Direct Take

Trigger a given Direct Take template. You must provide the *number* of the desired template.
You can also optionally show or hide the Direct Take icon.

Copyright © 2026 Vizrt

Take Template

This action takes a template by its type and variant (name) and which bus.

- **Template Type**: Selection of template type first to be used. For example, *Camera*, *Package*.
- **Variant**: Exact variant name derived from your *channel templates* belonging to the chosen template type.
- **Program/Preview**: Select the preferred way of inserting this template into the current running story
- **Show Icon**: Show/Hide the Template type icon on the Stream Deck device button.

Take Overlay Graphics

Take an overlay graphic on air.

- **Type**: Choose *Static Name* to always take the graphic named below, or *Bind to current Story* to take an overlay graphic belonging to the story that is currently on air. A key bound to the current story shows the name of the graphic it will take.
- **Name**: The name (slug) of the graphic. This field is shown for *Static Name* only.

Rehearsal Mode

Choose whether to enable/disable/toggle Rehearsal mode.

A Rehearsal Mode icon on the Stream Deck device displays *yellow* when Rehearsal Mode is activated, either from this action or from the Mosart GUI.

Skip Next

By choosing *Skip Type* in the drop-down menu, the user can skip either the next item or sub-item, in the rundown.

Unskip Next

By choosing *Skip Type* in the drop-down menu, the user can unskip (undo) a skipped item or sub-item, in the rundown.

Control Command

This action will execute a control command on Viz Mosart.

Choose the control command to execute from the drop down menu. You can add parameters in the text area in the form “Param1=val1”, separated by new line.

> **Tip**: For a full list of the supported control commands and their parameters, refer to the *Viz Mosart Administrator Guide*, in the *Appendix* section *Mosart Remote Control REST API*.

15.1.4 Status

The status of the current connected server is displayed.

Copyright © 2026 Vizrt

 GREEN: Connected

 YELLOW: Standby/Idle

 RED: Not connected

This key is disabled on the Stream Deck device, pressing performs no action.

 **Note**: When installing the Mosart Plugin, you may need to close and reopen the Stream Deck client application, to enable displaying of *status*.

Copyright © 2026 Vizrt

15.2 Vizrt SSO (Keycloak)

15.2.1 Working with Vizrt SSO (Keycloak)

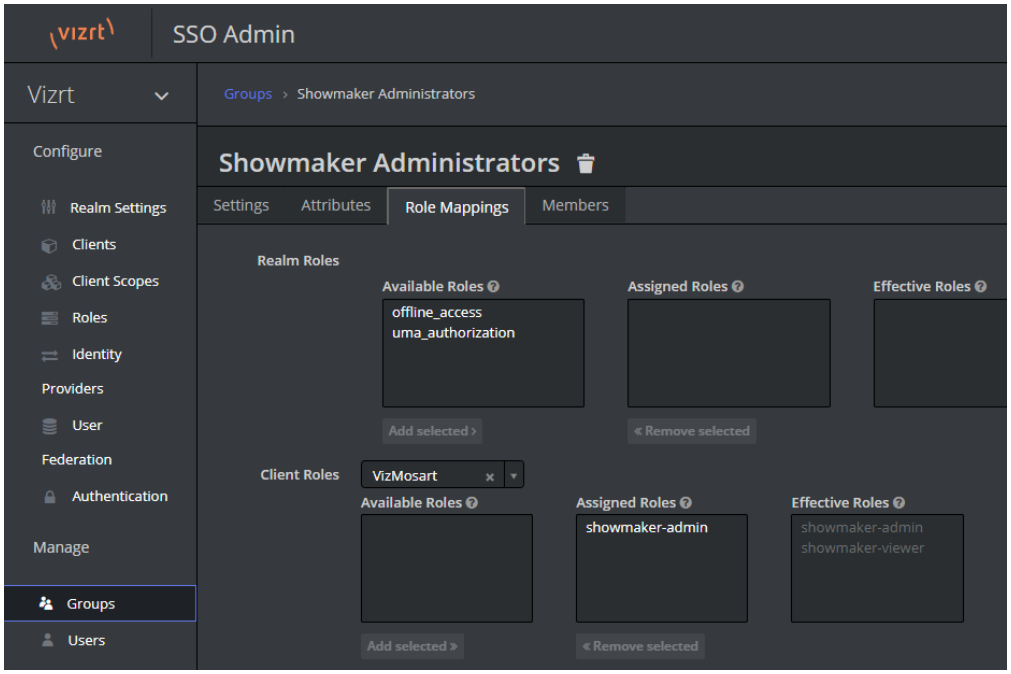
To connect the Mosart Web Applications to a Vizrt SSO (for example, when running together with a Viz One authenticated by SSO), perform the following steps:

1. Download the `keycloak-mosart-sso.json` file from the same folder that you got the installer on Vizrt's download site.
2. Log in to the Viz One SSO Admin pages, and go to the **Import** tab in the left menu.
3. Click **Select file** and pick the `keycloak-mosart-sso.json` file.
4. Review the options and when ready, click **Import**.
5. Go to the **Clients** tab, find and open the *VizMosart* client that was just imported.
6. Update the **Root URL** field with the fully-qualified domain name of the Mosart Web Applications server. The imported client arrives with the literal value `https://REPLACE.ME:55143`, so replace `REPLACE.ME` and keep the port that matches your setup: 55143 is the proxy's HTTPS port and 55142 its HTTP port. The client's **Base URL** is `/showmaker/`.
7. Manage users and groups to assign the Mosart Web Application roles as desired. The roles are defined by the selected **Client ID** *VizMosart*, as shown in the screenshot. The imported configuration adds four client roles, **showmaker-viewer**, **showmaker-admin**, **smartpanel-viewer** and **smartpanel-admin**, together with four example groups, **Showmaker Viewers**, **Showmaker Administrators**, **SmartPanel Viewers** and **SmartPanel Administrators**. Use the groups as-is, customize them, or replace them with your own.

Info:

- Groups do not directly grant permissions, they assign corresponding *roles* to all group members.
- Roles grant user permissions.
- *showmaker-admin* is a composite role that also grants *showmaker-viewer*, and *smartpanel-admin* likewise also grants *smartpanel-viewer*. The grant does not cross applications: a Showmaker administrator gets no Smart Panel permission, and a Smart Panel administrator gets none in Showmaker.

Copyright © 2026 Vizrt



Note: The imported configuration covers Showmaker and Smart Panel only. The Named Overlays Editor authorizes on **mosart-admin** and **mosart-viewer**, which are named after the capability rather than the app because Named Overlays and Smart Panel drive the same Viz Mosart Server endpoints. Neither role is in the file, so after importing it nobody is authorized in the Named Overlays Editor until you create those two roles on the **VizMosart** client and assign them.

8. Use the Configuration Tool to configure OIDC settings for the target server.
   The settings are on the **Server Configuration** page, in the **Authentication (OIDC)** section. That section appears only when HTTPS is enabled with **Certificates** set to **Custom certificate**. With **Automatic (built-in CA)** there is no Authentication section, and authentication is not enforced. **Manual proxy setup** always uses a custom certificate, so the section appears there as soon as HTTPS is on.
   a. **Issuer URL** is typically the Viz One server's base URL followed by `/auth/realms/Vizrt`.
   b. **Client ID** must be *VizMosart* when using the imported configuration.

Tip: The *Secret* key can be obtained from the Vizrt SSO Administration console.
- Select **Clients > Credentials** (tab).
- Copy the value of **Secret**.

Copyright © 2026 Vizrt