



Media Sequencer Release Notes

Version 5.9



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

2026/08/31

Contents

1	Media Sequencer 5.9.0.....	5
1.1	System Requirements	5
1.2	Silent Installation	7
1.3	Running the Media Sequencer	7
1.4	Removed Functionality	7
1.4.1	Legacy Element and Template Structures	8
1.5	Media Sequencer Core Enhancements and Fixes	8
1.6	New Plugins	11
1.7	Plugin Enhancements and Fixes	11
1.8	Known Issues	15
2	Documentation	16
3	Support.....	17

- [Media Sequencer 5.9.0](#)
 - [System Requirements](#)
 - [Silent Installation](#)
 - [Running the Media Sequencer](#)
 - [Removed Functionality](#)
 - [Media Sequencer Core Enhancements and Fixes](#)
 - [New Plugins](#)
 - [Plugin Enhancements and Fixes](#)
 - [Known Issues](#)
- [Documentation](#)
- [Support](#)

1 Media Sequencer 5.9.0

Release Date: 2026-08-27

These are the release notes for the Media Sequencer (MSE) version 5.9.0. This document describes the user-visible changes that have been done to the software since release 5.8.0.

Note:

Refer to comprehensive documentation of the Media Sequencer (MSE) in the *MSE Manual*. When the sequencer is running, this manual is available at http://localhost:8580/mse_manual.html. You can also access the manual by opening the local file `www/mse_manual/index.html` in the MSE install directory.

Since version 3.1, Media Sequencer is only available as a 64-bit software.

1.1 System Requirements

Software

Supported operating systems:

- **Server:** Windows Server 2016 or higher.
- **Workstation:**
 - Recommended: Windows 11.
 - Windows 10 (LTSC 21H2) is also supported.

Since Media Sequencer 5.9.0 is 64-bit software, it can only be run on 64-bit versions of Windows.

Operating systems that have reached their end of life cycle and are no longer officially supported by their manufacturers, are not supported for use with the MSE.

A Media Sequencer used with a Viz Pilot system needs to access the database which Viz Pilot uses. For Viz Pilot version 6 or older, this is an Oracle database, which previous Media Sequencers had to connect to directly.

While this configuration remains supported, Media Sequencer 4.0 and Viz Pilot 7 introduced the ability for the Media Sequencer to connect to the Pilot Data Server instead. The Pilot Data Server introduced in Viz Pilot 7 can in turn, run with either an Oracle backend or a Viz Graphic Hub backend.

When a Media Sequencer is set to connect directly to the Oracle database, an Oracle Database Client (Runtime or Administrator) is needed. Administrators must take care to install the 64-bit version of the Oracle Database Client.

Microsoft .Net

The Media Sequencer requires Microsoft .Net framework 4.7.2 (full) or newer to be installed on the machine. The installer notifies the user if .Net 4.7.2 is not found.

Licensing

As of version 5.4.0, Media Sequencer requires WIBU licensing. The WIBU CodeMeter runtime must be installed and configured, to make Media Sequencer licenses available. CodeMeter is installed as part of the new bundle installer.

Selecting which license container to use and which licenses to enable, is configured from the Media Sequencer launcher with elevated privileges. The launcher generates a license configuration file *licenses.json* in the Media Sequencer data directory. A Media Sequencer core license is required to launch. Additional licenses must be enabled for running the sequencer as a central Gateway, or for licensing the Viz Multiplay client.

As of version 5.5.0, Media Sequencer supports Viz License version 3 (movable licenses), in addition to the already supported Viz License version 2 introduced with 5.4.0. The launcher has been changed to allow selecting between version 2 and version 3 licenses, in the case where both are available.

As of version 5.8.0, Media Sequencer requires *MSE_PLAYOUT* license for playout and *MSE_FRAME_PREVIEW* license for Frame Rendering API.

The MSE can be configured to alert about license expiration in two ways, either via email (SMTP) or via Graphic Hub journal messages. This alert ability can be configured from the License Alert configuration, available from the launcher when the MSE is running.

For information about how to configure MSE licenses and license alerts, refer to the MSE WIBU licensing documentation (available from the Windows Start menu).

Hardware

System requirements for hosting the Media Sequencer:

Workload	Processor	Memory
Standard	2 cores / 4 threads	6 GB
Demanding	4 cores / 8 threads	12 GB

Running other resource-intensive software on the same machine is not recommended as it can slow down the Media Sequencer.

Virtualization

The Media Sequencer may be run inside a virtual machine. We recommend following the table above when allocating resources to the virtual machine. For time-critical operations we recommend using the Element Scheduling REST API to trigger frame accurate playout.

Applications and Components Recommended with this Release

- Viz Trio 4.6.1 or higher
- Viz Multiplay 3.3.0 or higher
- Viz Pilot 8.9.5 or higher
- Viz Pilot Edge 3.6.0 or higher
- Template Builder 3.6.0 or higher
- Pilot Data Server 9.6.0 or higher
- Preview Server 4.7.1 or higher (Preview Server 5.0.0 requires MSE Frame Rendering API)
- Graphic Hub 3.9.3 or higher
- Graphic Hub REST 2.9.3 or higher
- Graphic Hub ImEx Agent 2.2.0 or higher
- Viz Engine and Viz Artist 5.5.2 or higher
- Viz One 7.7.0 or higher
- Viz Mosart 5.16.1 or higher

The Media Sequencer has been designed to have backward compatibility with older versions of these components. We work hard to keep the Media Sequencer backward compatible, however, for some older versions of the Delphi-based clients (for example, *Director* and *Viz Trio*), we highly recommend upgrading these clients to a more recent version.

1.2 Silent Installation

Silent installation allows the Media Sequencer to be installed without user interaction.

To perform silent installation, open a command shell as an administrator and run the Media Sequencer bundle installer with the `-s` or `--silent` option.

It is also possible to select features to be installed from the command line.

To get all options available, run the Media Sequencer bundle installer with the `--help` option.

Example command:

```
MediaSequencer.BundleInstaller-5.x.x.xxxxx.exe -s
```

Example command to install Media Sequencer with Gateway controller:

```
MediaSequencer.BundleInstaller-5.x.x.xxxxx.exe -s --mse-gw-controller
```

A license configuration file *licenses.json* in the Media Sequencer data directory (*C:\ProgramData\Vizrt\Media Sequencer*) is required to run Media Sequencer. If *licenses.json* is missing, the installer will prompt the user to create it using the Media Sequencer launcher. Note that license configuration cannot be changed using the non-admin Media Sequencer launcher.

To avoid the automatic license configuration prompt before running silent installation, copy *licenses.json* from another successful Media Sequencer installation to the Media Sequencer data directory. See the [License](#) section for more information.

1.3 Running the Media Sequencer

It is highly recommended that the Media Sequencer is run as a system service, rather than in a console. This is the default behavior after installing.

To start the Media Sequencer, licenses must be configured. This can be done from the launcher with elevated privileges. For more information about licensing see the [License](#) section, and the MSE WIBU licensing documentation (available from the Windows start menu).

The Media Sequencer can be started and stopped using the Media Sequencer launcher. You can do both with and without elevated privileges. In order to start and stop the Media Sequencer without elevated privileges, open the shortcut **Media Sequencer (non admin)** available from the Windows Start menu. Note that it is *not* possible to configure any settings from the launcher without elevated privileges.

1.4 Removed Functionality

Deprecated Plugins

Plugins are sometimes deprecated with a new release of the MSE. These plugins will no longer be available in the next release. You need to migrate away from using them.

The following plugins are deprecated in this release:

- None.

1.4.1 Legacy Element and Template Structures

The template structures which are not compliant with Four-Layer template structures are now deprecated. These templates are referenced from element nodes using "template", "templates", or "template_path" indicators.

Clients should instead use Four-Layer template structures which is referenced from element nodes using "master_template" entry.

Deprecation reference ids: 590-01 to 590-06.

(MSE-9833)

1.5 Media Sequencer Core Enhancements and Fixes

Enhanced Trusted Service Validation for Single Sign-on Authentication

Trusted services can now be configured using host:port or URL prefixes. This allows users to specify requirements for SSO providers such as schemes and paths. The Media Sequencer performs SSO authentication for requests that match the configured trusted services.

(MSE-9543)

Input Validation for Single Sign-on Configuration in the Media Sequencer Launcher

The Media Sequencer Launcher now validates the Single Sign-on (SSO) configuration entered in its settings, including the issuer URL, client ID, client secret, and scope. Invalid entries are highlighted and an error message is shown, and the SSO settings cannot be saved until all errors are resolved.

(MSE-9751)

Improved Single Sign-on Logging

Single Sign-on (SSO) log messages are now emitted under the logger of the handler that requested the access token, for example playable_cache, so SSO logging can be enabled by raising log level in that handler settings rather than the root log level. SSO error messages also include more context, such as the endpoint that failed and the reason.

(MSE-10067)

Support Atom Document Monitor over Websocket

Atom Document Monitor now supports subscribing to STOMP messages over WebSocket, both secured and non-secure connections.

(MSE-9426, MSE-9573)

Improved error message in Atom Document Monitor when monitoring STOMP URLs without port numbers

Before starting to monitor, Atom Document Monitor validates the STOMP URL. If the STOMP URL is invalid, it logs a warning message and does not begin STOMP monitoring.

(MSE-9864)

Support Single Sign-on Authentication in Atom Document Monitor

Atom Document Monitor now supports monitoring SSO-protected resources via both REST and STOMP channel.

(MSE-9465)

Optimize Viz Commands when Generating Snapshots in Frame Rendering API

The MSE Frame Snapshot API provided a new 'optimize_tl_state_o' parameter to enable optimization which improves response time when creating a snapshot for a transition-logic element with a large number of toggle layers in state 'O'.

The optimization assumes that the layers in state 'O' are invisible and simply sets the layers to 'O' without loading any geometries.

This optimization is disabled by default.

(MSE-9824, MSE-9836)

Fixed Read Access Violation when Logging Unexpected Field in List Payload

Fixed an issue where logging a message about an unexpected field in a list payload could cause a read access violation, if the list field name was a nullptr. The log now correctly reports the name of the corresponding field.

(MSE-9834)

Fixed an issue where MSE would fail to parse a URL with query parameters but an empty path

(MSE-9869)

Update WIBU CodeMeter Runtime to version 8.40a

(MSE-9923)

Updated CodeMeter WebAdmin URL in the Media Sequencer Launcher

The Media Sequencer Launcher now opens the CodeMeter WebAdmin directly instead of going through a redirect.

(MSE-9976)

Bundle Installer Uses a Less Generic Name

The bundle installer's log is now written under `%LOCALAPPDATA%\Vizrt\MediaSequencer.BundleInstaller` instead of the more generic `Vizrt\BundleInstaller`.

(MSE-9989)

Support initializing and cleaning-up material_definition resource type

The `material_definition` is a resource type with mediatype defined as `application/vnd.vizrt.viz.material_definition`. It is a new resource type for the new Viz renderer pipeline and is equivalent to the 'material' resource type for classic scenes. Media Sequencer can load the 'material_definition' resources on initialization and unload them on clean-up.

(MSE-9930)

Self-connect prevention for connections to Viz Engine

Media Sequencer now prevents self-connection scenarios when establishing TCP connections to Viz Engine located on the same machine. When the operating system assigns a local ephemeral port number that matches the target Viz Engine port, the connection is detected as a self-connection and automatically retried with a different port.

This addresses a rare but possible situation where the operating system assigns a local port from the ephemeral port range that conflicts with Viz Engine ports. For example, port 50007 is used by Viz Engine as the MUX isolated port for frame snapshot preview. Without this prevention mechanism, such port conflicts could result in the Media Sequencer attempting to connect to itself rather than the intended Viz Engine instance.

(MSE-9926)

Fixed possible MSE crash on exit if Director is running and playable_cache is enabled

(MSE-9977)

Improve MSE Cleanup Viz Commands

Added `POOLS CLEANUP` in viz cleanup logic.

(MSE-9588)

Fixed the performance issue with transition logic state serializing

(MSE-9972)

Fixed Transition Logic State Serialization Failing on Large Field Values

When serializing transition logic element state, the Media Sequencer applies pattern-matching rules that strip the authentication parameter from resource URLs, so that the same resource is recognized when comparing element state. For fields containing a large value, applying these rules could become very slow and fail with an error, leaving the field state unserialized. The rules have been improved so that large field values are processed efficiently and no longer cause this failure.

(MSE-9747)

Fixed an issue preventing multiple Viz superchannel actors from checking the connection simultaneously.

(MSE-10031)

Support Frame Snapshot Request with Format Field (Adaptive Graphics Support)

The Frame Snapshot API now supports requesting snapshots with a specific format preset (e.g., using the `<format>` field in a snapshot request) to define the output resolution and aspect ratio.

When a requested scene's camera aspect is configured as aspect format (`ASPECT_FORMAT`), Media Sequencer queries the available layout format presets from the Viz Engine using `SCENE*<scene>*FORMAT*DATA GET` . It then determines the correct dimensions or aspect ratio to calculate the final snapshot size. If the requested format is not found or has its format size disabled (`use_format_size=0`), the sequencer automatically falls back to either the `default` format preset or the global renderer output configuration.

Additionally, Media Sequencer always sets the requested format on the renderer using `RENDERER*FORMAT*REQUESTED_FORMAT SET` with the requested format name or with `default` if none was specified.

(MSE-10088)

Authenticode Signing of Installed Files

All executables and libraries delivered by the Media Sequencer, including the launcher and the installers themselves, are now Authenticode signed. This allows Windows to verify the authenticity and integrity of the files, and reduces the likelihood of anti-virus software flagging or blocking the Media Sequencer.

(MSE-9794)

Fixed Incorrectly Displayed WIBU License in Media Sequencer Launcher

When multiple versions of a license were available on a license server, the main window of the Media Sequencer Launcher could show an incorrect 'Count' and 'Expiration date' compared to what was selected in the Media Sequencer Settings. The launcher now retrieves real-time license information from the license server and, when the Media Sequencer is running, synchronizes with the active licenses reported by the sequencer to display the correct

license details. When the launcher is not connected to a running Media Sequencer and cannot resolve which of several same-named licenses is in use, the license is marked with a "*" suffix, its count and expiration are shown as unavailable, and a warning directs the user to connect to a Media Sequencer for real-time information or to consult Settings > Viz Licenses for details about the possible license candidates.

(MSE-9522)

Fixed "Remove Deleted Elements" Option Having No Effect in Pilot Database Configuration

In the Pilot Database configuration (PilotDbConfig), enabling the "Remove deleted elements" option for the Pilot Data Server backend had no effect. The option is now applied correctly, so that elements removed from the Pilot Data Server are automatically removed from the Media Sequencer as configured.

(MSE-9702)

1.6 New Plugins

1.7 Plugin Enhancements and Fixes

as_run_log

- Support as_run_log when running in superchannel arm/fire mode. The as_run_log is triggered by 'superchannel_manager' once a command affects the playout. The as_run_log is not triggered when an element is 'arm' or 'unarm'. Once the armed element is 'fire' or 'fire_all', the as_run_log is triggered for the element (graphic or video) as command 'take' (MSE-9366).

atom_document_monitor

- Improve content resolving mechanism for STOMP updates containing empty bodies. The atom_document_monitor now better handles scenarios where multiple updates for a single resource arrive with empty bodies, preventing endless fetching and ensuring reliable content resolution (MSE-9846).

channel_state

- Expose a 'based_on_element_id' attribute in channel state, giving the atom id of the Element Collection Entry that an on-air or armed element is based on, alongside the existing 'based_on' URI. This remains available even when an element is deleted between arm and fire, by using a snapshot taken at arm time (MSE-10133).

gateway

- Fixed Gateway stops responding when a failover pair is activated (MSE-9897).
- Improved handling of query parameters for MOS 4.0 WebSocket connection URIs. The Gateway now automatically adds the required 'mosID' and 'channel' parameters to the configured WebSocket URI, so the mosID no longer has to be entered manually in the Gateway configuration. Parameters are URL-encoded, any parameter already present in the URI is kept instead of being overwritten (a warning is logged when this happens), and URI fragments are preserved (MSE-9911).

http_server

- Support creating videowall preset elements through REST API. Clients can send POST requests to Element Collection or Hierarchy Collection to create videowall preset elements which can be recognized by Multiplay (MSE-9684).
- Provide a correct value of 'main_channel_port' field in Videowall payload (MSE-9755).
- Expose Graphic Hub information for videowall scenes in VDF payload. The Video Wall Scene Collection Entry now includes a VDF content payload with the scene's 'sceneName', 'sceneID', 'scenePath' and 'description', together with its atom link ('atomURL') to Graphic Hub (MSE-9656).
- Default model resource will transform uri in choice src attribute base on request uri (MSE-9853).
- Fixed a possible unhandled exception when sending a POST or PATCH request to a Hierarchy Collection while also removing the collection node (MSE-9891, MSE-9896).
- Refined the process for creating a new directory, show or playlist by Directory resource. Clients can now send a POST request to a Directory Collection resource with an Atom Entry in the request body; only the atom title and category are required. The atom category specifies a resource type to create the atom title specifies a name of the created resource. Optionally, clients can specify the directory name using the 'slug' header (MSE-9906, MSE-9907, MSE-9908).
- The directory root at '/directory' is reserved exclusively for 'shows' and 'playlists' directories. Media Sequencer returns a 400 Bad Request response to POST requests attempting to create Directory Entry resources other than 'shows' or 'playlists' directories (MSE-9970).
- Fixed an issue where Media Sequencer did not consider the base URI when processing URIs in asset documents. This could cause Media Sequencer instances running behind proxy servers to incorrectly interpret resource URIs, such as model URIs (MSE-9951).
- Fixed an issue with the REST API for scene import from BGFX mastertemplate, where some markup could be stripped from the VDF model stored in the new sceneinfos for the template (MSE-9971).
- Update template entry serialization to use templatedescription when present. This will check 'templatedescription' attribute on the mastertemplate node first, then fall back to the attribute on the first sceneinfo element of the template (MSE-10059).
- Added a MOS playout workflow guide with examples to the REST API documentation, describing how to set up and control a MOS-driven playout workflow using the REST API (MSE-10130, MSE-10131).
- Improved MOS lookup for directory resources so that it returns resources consistent with those discoverable under the '/directory' tree. For backward compatibility, the lookup falls back to storage paths when the resolved directory resource is not a descendant of '/directory' (MSE-10152).
- Allow Profile Command Resource to accept a MOS Resource URI directly as a playout target. This simplifies the MOS playout workflow by letting clients reference an element by its ncsID, roID, storyID and itemID without first resolving it to an Element Collection Entry URI (MSE-10132).
- The HTTP server now applies explicit, configurable limits to incoming requests. Previously an implicit 1 MB body limit applied, and exceeding it (or exceeding the header limit) closed the connection with no response, so the client only saw a dropped connection. Request bodies are now accepted up to a configurable limit, controlled by the new 'max_request_body_size_bytes' handler setting (default 10 MB; set to 0 to disable); larger requests are rejected with "413 Payload Too Large", for both Content-Length and chunked bodies. A request whose request line and headers together exceed 8192 bytes is now rejected with "431 Request Header Fields Too Large" rather than silently closed; this header limit is not configurable (MSE-10181).
- Invalid videowalls are no longer silently omitted from the videowall feed. Media Sequencer logs a warning when it skips an invalid entry and, instead of dropping a wall or replying 404 for a bad 'main_channel_port', returns the wall with an error annotation on the 'main_channel_port' field so clients can surface the problem (MSE-9893).

- Support resolving MOS objects synchronously through Playable Cache. When the Pilot backend is configured as Playable Cache, POSTing a MOS item that supplies an 'element_uri' resolves the element synchronously, replying '201 Created' on success or '400 Bad Request' on failure (previously '500 Internal Server Error'). When 'element_uri' is missing, Media Sequencer logs a warning and falls back to the legacy vcp_service behavior, creating a pending element ('202 Accepted') (MSE-9892).
- Profile Command Resource now accepts an optional `channel` query parameter to target a specific channel on all element commands. The `read` command is the exception: it always previews on the dedicated preview channel, so specifying `channel` is rejected with a 400 Bad Request. See the REST API documentation for the full list of supported commands (MSE-9763, MSE-9811).
- Profile Command Resource now accepts a Playout Document (media type `application/vnd.vizrt.playout+xml`) as the body of a POST (`arm`, `take`, `out`), providing additional context for a superchannel arm/fire videowall, such as the preset transition and content transitions to use. A new Playout Document Specification page with examples has been added to the REST API documentation (MSE-9880, MSE-9957, MSE-9992).

peptalk

- Added new VDOM Binding API for subscribing changes on element path, children, attributes, text, and has-children state. For further information, refer to the following sections in the Media Sequencer Manual: Subscribe-element-change, Subscribe-child-change, and Unsubscribe commands (MSE-9861).
- Fixed invalid URI replied by Peptalk uri command for a Hierarchy Collection resource representing a show or playlist (MSE-9914).

playable_cache

- Prevent undefined behavior if the issuer URL setting for Single sign-on (SSO) is empty (MSE-9746).
- Support Pilot system's external_id in Playable Cache. The `external_id_uri` field must be defined such that the external ID template can be resolved from URI by Playable Cache (MSE-9538).
- Support `live-update` auto-convertible document provided inline in the Pilot template resource (MSE-9749).
- Support multiple STOMP monitor URLs in Atom document. MSE selects the URL with the highest precedence scheme. The precedence order from highest to lowest is: 'stomp+wss', 'stomp+ws', and 'stomp' (MSE-9840).
- Prevent unnecessary re-subscription when the STOMP monitor link changes without pointing to a different endpoint. Pilot Data Server 9.4.0 provides a per-resource STOMP monitor link that includes a timestamp of the resource state the client has last seen, so the link changes each time the resource is received again. The Media Sequencer now compares only the host and port of the monitor link and re-subscribes only when the endpoint actually changes (MSE-9845).
- Fixed an issue where creating a template element from model with material_definition fielddef is not recognized and skipped (MSE-9929).
- Fixed an issue where some markup in pilot template VDF models were stripped before being stored in cache (MSE-9971).
- Removed legacy "templates" references from sceneinfos generated by playable_cache (MSE-9987).
- Made URL-encoded Playable Cache references easier to work with. Log messages during Playable Cache resolution now show the decoded resource URI instead of its percent-encoded form, and the VDOM configuration web app (vdomconfig) provides a new "Edit ref" tab for viewing and editing URL-encoded Playable Cache references (MSE-9954).

- Reject invalid Playable Cache references immediately instead of silently mishandling them. A reference with an incorrect path — for example an extra path segment before the encoded URL, or an unescaped trailing slash — was previously accepted and only caused problems later during cache cleanup, making the cause hard to diagnose. Such references now fail right away with an error status describing the invalid URL. The VDOM configuration web app (vdomconfig) validates reference encoding accordingly when creating or editing Playable Cache references (MSE-9966).
- Expose the editor URL of Pilot elements resolved by Playable Cache. When a Pilot asset provides an editor link, the resolved element now carries an 'editor' attribute with that URL, allowing clients to open the element in its editor (MSE-10086).
- Support resolving a filled preset without an explicit 'preset_name'. When it is omitted, Playable Cache uses the default 'preset_name' from the template model; when no preset name is defined at all, the request is resolved as a quick preset if the template supports it and the payload provides a 'preset_layout' (MSE-9895).

superchannel_manager

- Fixed an issue in which superchannel_manager could resolve the incorrect background scene when playing a transition logic element with an alternative concept defined. This can cause superchannel_manager to wait for the wrong scene to transition, resulting in a timeout error and delay in the next taking element (MSE-9851).
- Select preset plugin command variants by preset plugin capability. Media Sequencer stores the preset plugin capabilities in the videowall node under the profile and exposes them as a 'capabilities' field in the videowall REST representation. When superchannel_manager sends commands to the preset plugin, it picks the command variant according to these capabilities (MSE-9878, MSE-9913, MSE-10158).
- Media Sequencer now lets clients control the preset transition and the content transitions per element when playing out to a superchannel arm/fire videowall. To specify these preferences, POST to Profile Command Resource (`arm` , `take` , `out`) with a Playout Document as the body. The document specifies the target element, an optional preset transition (`transition` or `cut` , with an optional duration), and one or more content transitions (with channel and optional duration). Media Sequencer checks that the target preset plugin capabilities supports the requested transitions and rejects the request with a 400 Bad Request when it does not. A preset transition `cut` does not need content transition capability (MSE-9771, MSE-9809, MSE-9900, MSE-9922).

vcp_service

- Support Single Sign-on (SSO) authentication when fetching and monitoring resources from Pilot Data Server (MSE-9463, MSE-9812).
- Fixed a possible crash when disabling the vcp_service actor while it waits for HTTP responses (MSE-9912).
- Fixed changelog monitoring stopping when a concept change event was received from the Pilot Data Server. Such an event was treated as an unknown resource type and stopped the monitoring, so the Media Sequencer no longer received live updates. Concept change events are now ignored and monitoring continues (MSE-9754).

viz

- Prevent infinite loop in MSE when retrieving large model XML from engine. This could happen when users use the "scene model retrieval" REST API to retrieve a model (MSE-10068).
- Fixed fill view mode returning a key-composited image instead of opaque fill when requesting PNG snapshots via "frame snapshot command" REST API (MSE-10070).

1.8 Known Issues

This section describes issues that have been reported but not resolved.

Cannot Configure WIBU SmartBind Loose Licenses in MSE Launcher

When activating a Software Container on a virtual machine or in a cloud environment via the license portal, additional manual configuration is required (MSE-9666).

Workaround Steps

1. Assign the required licenses in **MSE Launcher**.
2. Close the **MSE Launcher** application.
3. Navigate to the following file location: `C:\ProgramDataVizrtMedia Sequencerlicenses.json`.
4. Open the `licenses.json` file in a text editor **with administrative privileges**.
5. Locate the relevant license container and change its `"location"` value from `"0"` to `"1"`.

Note:

Editing the file requires the text editor to be run as an administrator.

Issue with "Out" Command when Performing Frame Accurate Payout of Transition Logic with Video

If you do frame accurate `out` and then two `take` operations of transition logic elements with video, the second video will not start from the beginning of the clip (MSE-7046).

MSE may Report the Main Thread Slow on the Re-initialization Process

MSE may report the main thread to be slow while it is re-initializing. This issue typically occurs when the Viz Pilot database is at a remote location.

Inaccurate Video Availability Progress if the Video Clip is Deleted from a Publishing Point

Availability progress of a video clip is not immediately reset to zero when the related video clip is deleted from a publishing point. This is due to the limitation of the Viz One API feedback (ME-664).

Ordering of Fields in Payloads and Models Not Respected

Media Sequencer does not respect ordering of fields and fielddefs in VDF payloads and models, meaning that for example, a payload inserted using the REST interface may have the fields in a different order when later obtained from the REST interface. This has no consequence for payout, but may cause unexpected reordering of fields and fielddefs in software used to display or edit payloads or models (MSE-6565).

Superchannels May Become Stuck in Transitioning State when Arm/Fire Non-Existing Video Clips

When operating in superchannel arm/fire mode, the Media Sequencer requires feedback from the Viz Engine to ensure that the Viz Engine has finished transitioning between playing and pending subchannels before sending next commands; otherwise, the next commands may affect the incorrect subchannel. The Media Sequencer holds any new commands during transitioning until it receives transition completed feedback from the Viz Engine. When a user arms a non-existent video clip, the Viz Engine does not notify the Media Sequencer that the clip is not available. If the user continues to fire this non-existent video clip, the Media Sequencer will stop executing new commands until the `viz_feedback_timeout_seconds` is reached. This `viz_feedback_timeout_seconds` can be configured in `superchannel_manager` handler settings (MSE-8414).

2 Documentation

Documentation for the Media Sequencer is available in the bundle installer on the [Vizrt FTP](#).

Once installed, documentation is available within the MSE, see the note above.

3 Support

Support is available at the [Vizrt Support Portal](#).