AI Keyer User Guide

Version 1.1

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 Introduction

The Vizrt AI Keyer separates focused people from the background to enable AR workflows without the need of a green screen.

	Virtual Studio
	Live input
	Matte result of the AI Keyer

	Composition result with the AI Keyer

1.1 Related Documents

- Viz Engine
- AI Terminal

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

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

1.2 Feedback and Suggestions

We encourage feedback on our products and documentation. Please contact your local Vizrt customer support team at www.vizrt.com.

1.3 System Overview

The Vizrt AI Keyer is a feature configured and controlled by the *Vizrt AI Terminal*. It integrates with *Viz Engine* to key one or multiple talents using AI. Viz Engine can then be configured to use the key data to put graphics in front or behind the keyed talents.

> **Note:** AI Keyer needs to run on the same computer as Viz Engine.

1.4 System Requirements

This page explains the recommended specifications for running the AI Keyer.

1.4.1 General

OS	Windows 10 (64-bit)
	Windows 11
Browser	Google Chrome
	Firefox
	Microsoft Edge

1.4.2 Hardware

	Keyer Standard 1.0	Keyer Standard 1.1	Keyer Studio 1.1
Box	HP Z4 G5	HP Z8 G5	HP Z8 G5
Graphics Card	A single NVIDIA RTX 6000 Ada Lovelace running the Viz Engine plus AI Keyer.	A single NVIDIA RTX 6000 Ada Lovelace running the Viz Engine plus AI Keyer.	A **dedicated** NVIDIA RTX 5000 Ada Lovelace running only the AI Keyer.

Note: The performance is strongly influenced by the complexity of the scene loaded in *Viz Engine*.

Note: The load can be distributed among two graphic cards, one running the Viz Engine and one running the AI Keyer.

Note: The AI Keyer also supports the follow up cards from the NVIDIA Blackwell generation.

1.4.3 Software

- Viz Engine (5.5 or later)

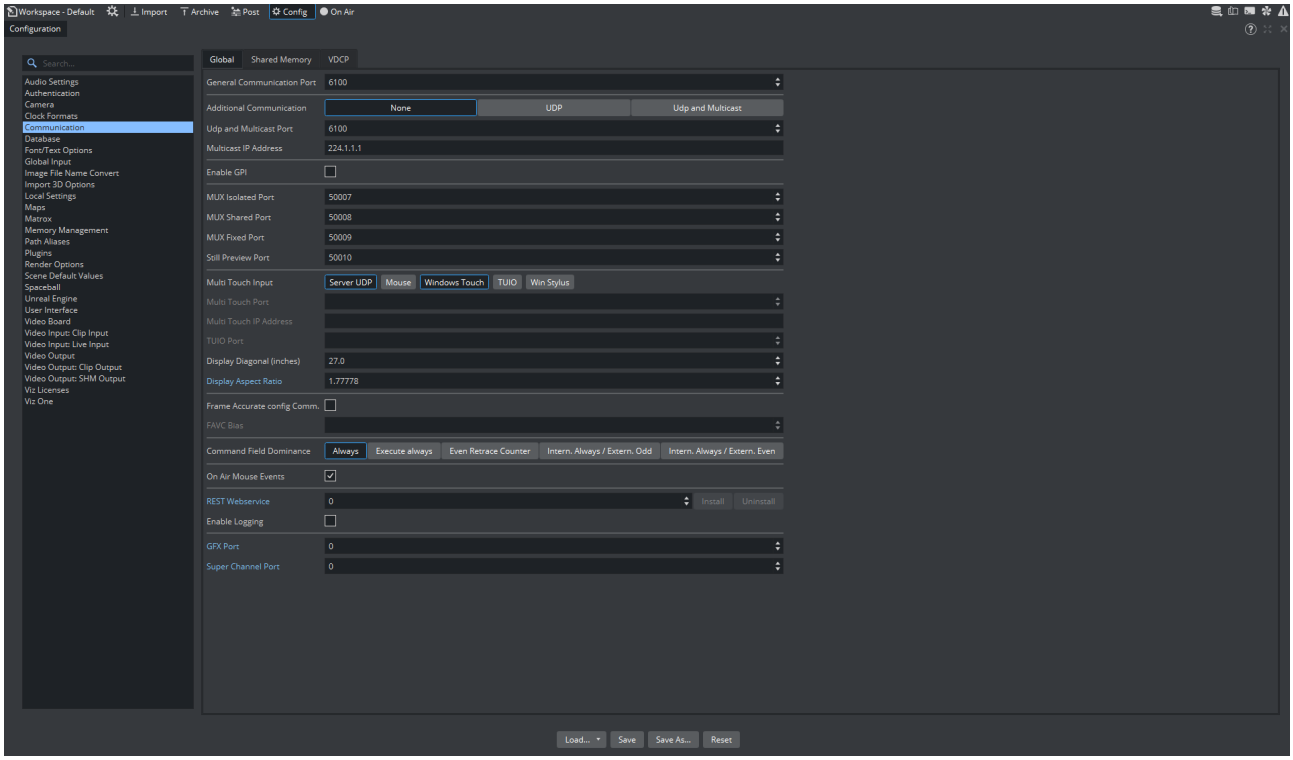
2 Installation

2.1 Branded Installer

1. Double click *Viz_AI_Tools-*.exe*.
2. Select **Requirements**, **AI Terminal**, and **AI Keyer**.
3. Click **INSTALL** and follow the instructions.

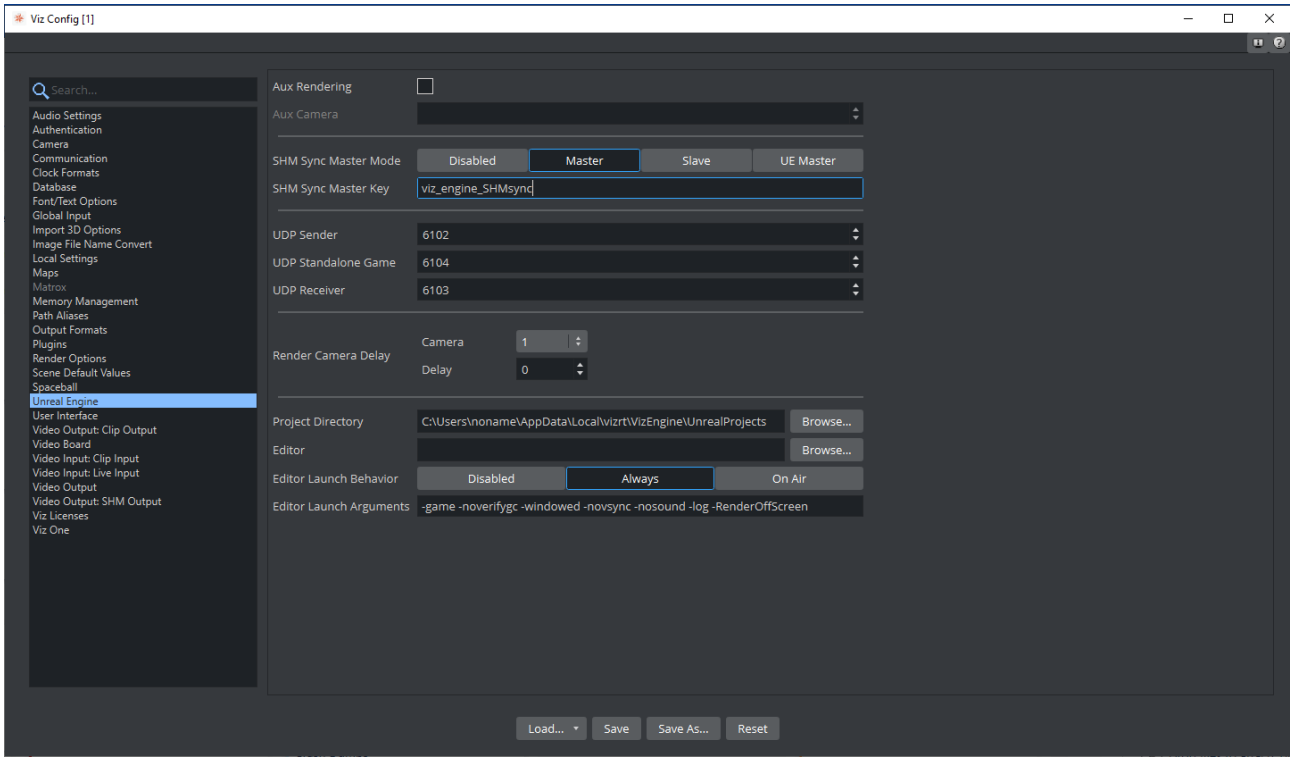
3 Viz Engine Configuration

3.1 Communication



Make sure to remember the General Communication Port. This number is required during the Tool Configuration.

3.2 Unreal Engine



3.2.1 SHM Sync Master Mode

Set to *Master*.

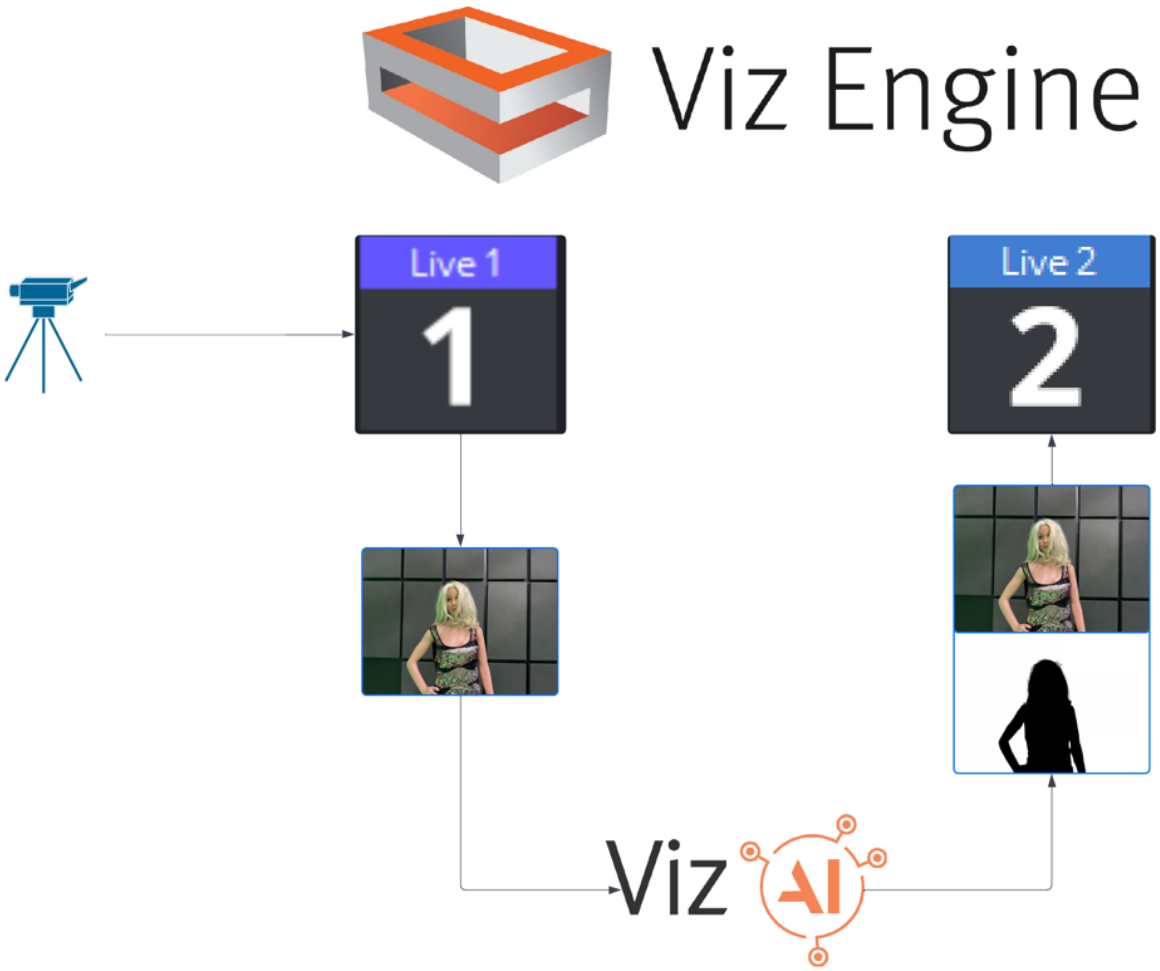
3.2.2 SHM Sync Master Key

Specify a non-empty name.

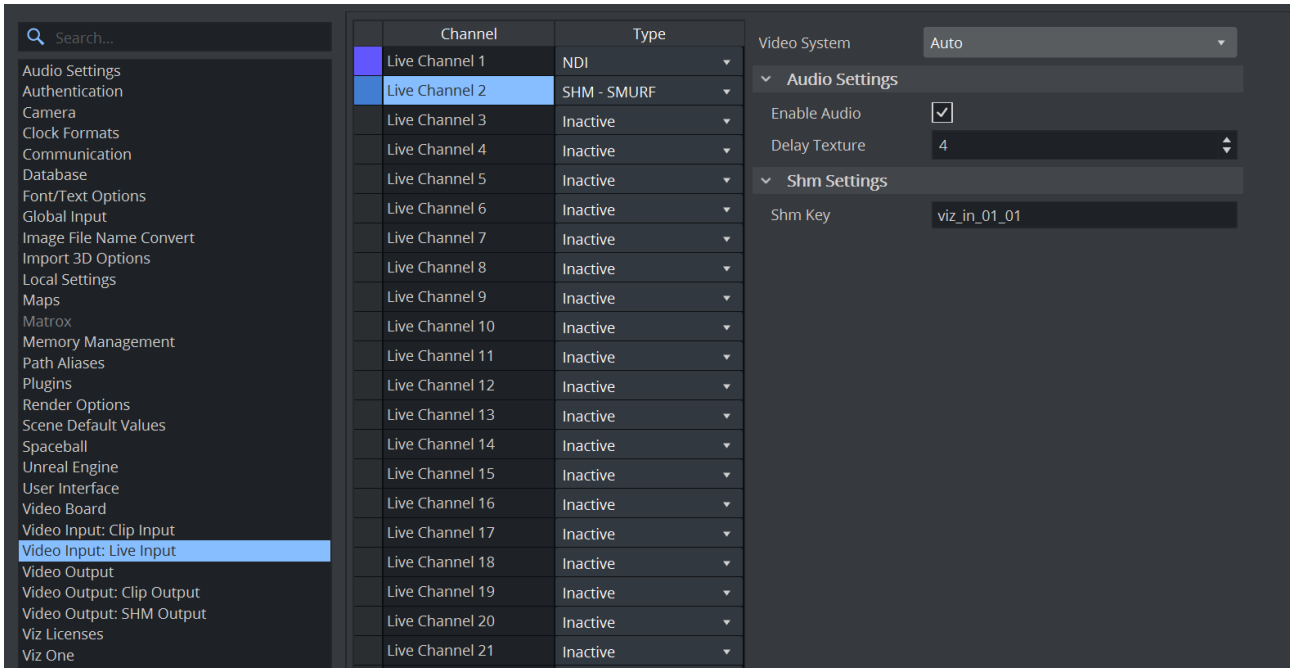
3.3 Video Input: Live Input

Enable at least one input source, that is the video of which foreground is being keyed. This is the source we configure later to be sent to the Vizrt AI Keyer. Then, enable a Live Channel as **SHM-SMURF**, which we configure later to contain the keyed result. Remember the Channel numbers for both input configurations. This is required during the Tool Configuration and Scene Design. Below an example with Live 1 as general video input channel and Live 2 as SHM - SMURF containing the keyed foreground signal.

Information: The AI Keyer also supports a Clip Channel as input. The channel receiving the AI Keyer output, however, is always a Live Channel.

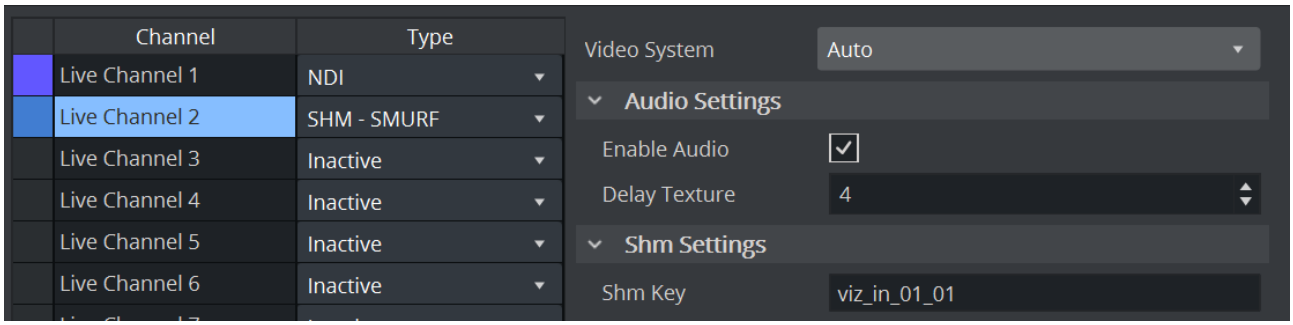


Live Channels can be enabled in the *Video Input: Live Input* section of the Engine's *Config* window:

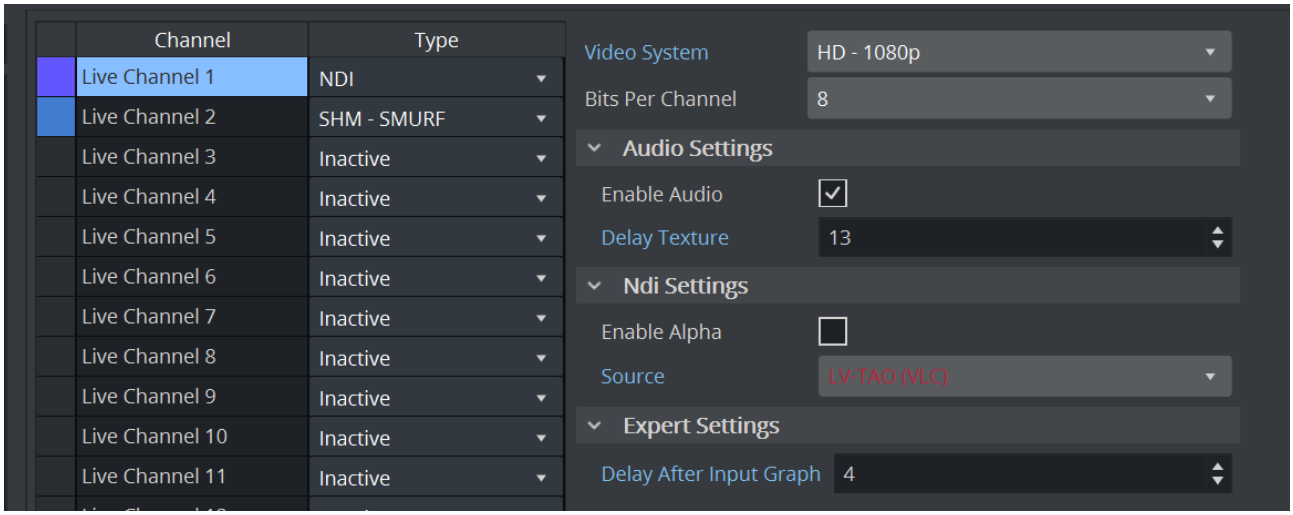


3.3.1 Texture Delays

When enabling a Live Channel as **SHM-SMURF** source, the *Delay Texture* value is automatically set to 4:



To sync the two inputs, the live input source channel must also be delayed (by the same value) using the *Delay After Input Graph* option:



4 Viz Engine Scene Design

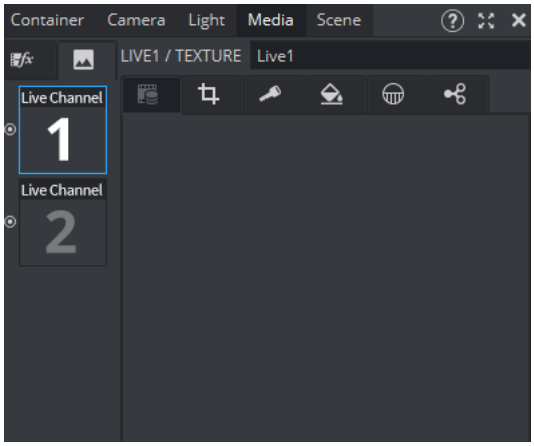
This section describes the steps required to design a Viz Engine scene that can use the AI Keyer capability.

Note: The AI Keyer requires a *Precision Keyer* or a *AI Keyer Greenscreen-less* license. Make sure to have these enabled in the **Viz Engine Settings > Viz Licenses** panel.

4.1 Media Tab

4.1.1 Input Sources

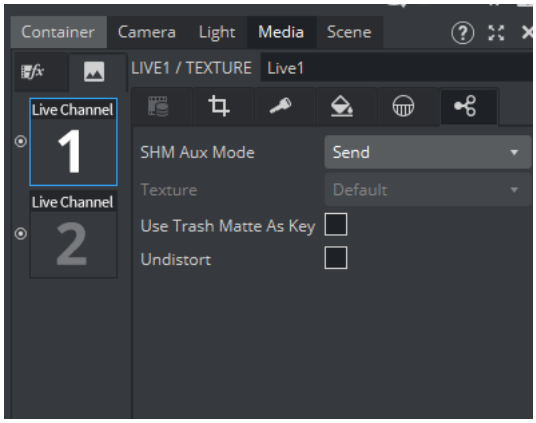
Drag the configured video input source and the shared memory SHM - SMURF source into the media Textures.



In this guide, we assume that Live Channel 1 is the live video input source and Live Channel 2 is the SHM - SMURF source, as shown in the Viz Engine Configuration section.

4.1.2 Input Source to AI Keyer Aux Setting

Make sure to set the *SHM Aux Mode* of the source for which AI keying is required to *Send*. This source is now also available as input source for the AI Keyer tool.



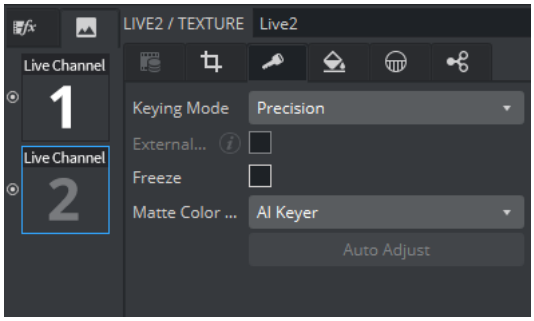
The *SHM Aux Mode* can be set in the **Media > Aux** tab of the video input source for which we want AI keying. In our example, this is Live Channel 1.

Note: The Input Source needs to be added to the Textures. If you cannot change the SHM Aux Mode, the Channel is likely configured as DVE. Right click on it and select **Set to Texture**.

4.1.3 SHM - SMURF Input Source Keyer Setting

Configure the ***Media > Keyer*** tab of the Live Channel configured as SHM - SMURF as follows:

- Set the **Keying Mode** to *Precision* - **if** you want to set the edge/blur/light wrap properties of Precision Keyer on the AI Keyer.
- Set the **Keying Mode** to *Linear* - otherwise.
- Set the **Matte Color Mode** to *AI Keyer*.



In our example, the SHM - SMURF configured Live input is Live Channel 2.

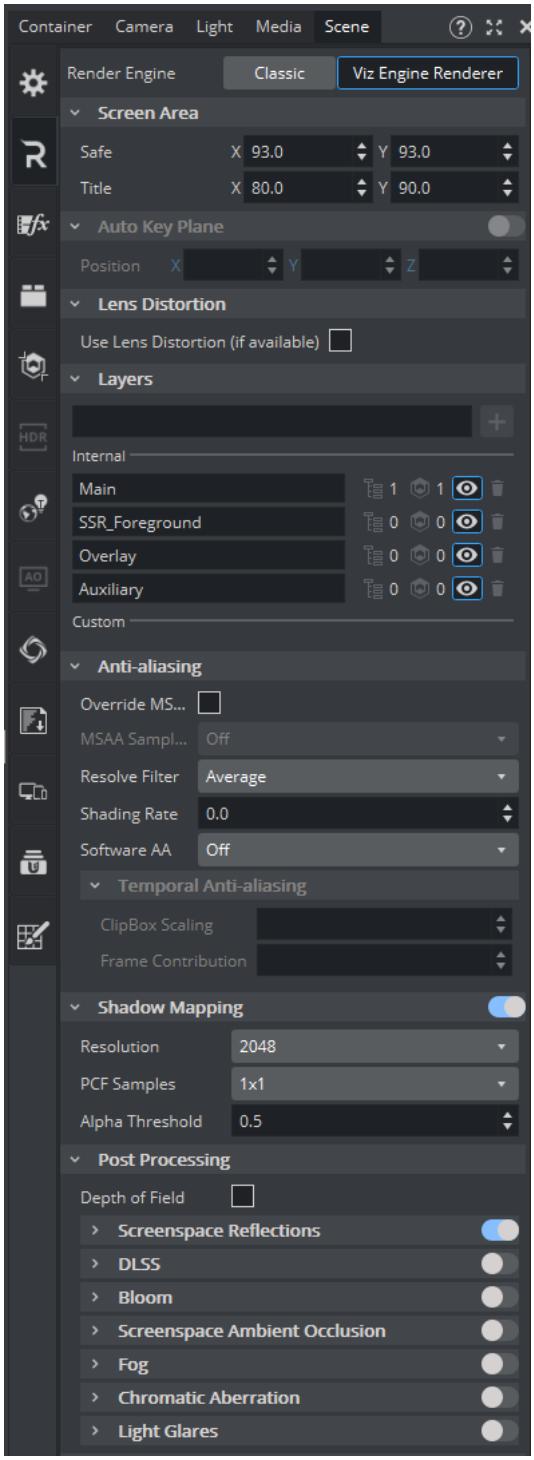
Note: The Input Source needs to be added to the Textures. If you cannot change the SHM Aux Mode, the Channel is likely configured as DVE. Right click on it and select **Set to Texture**.

Note: The edge/blur/light wrap properties of Precision Keyer can be applied also on AI Keyer. See the *Viz Artist User Guide* for a detailed explanation.

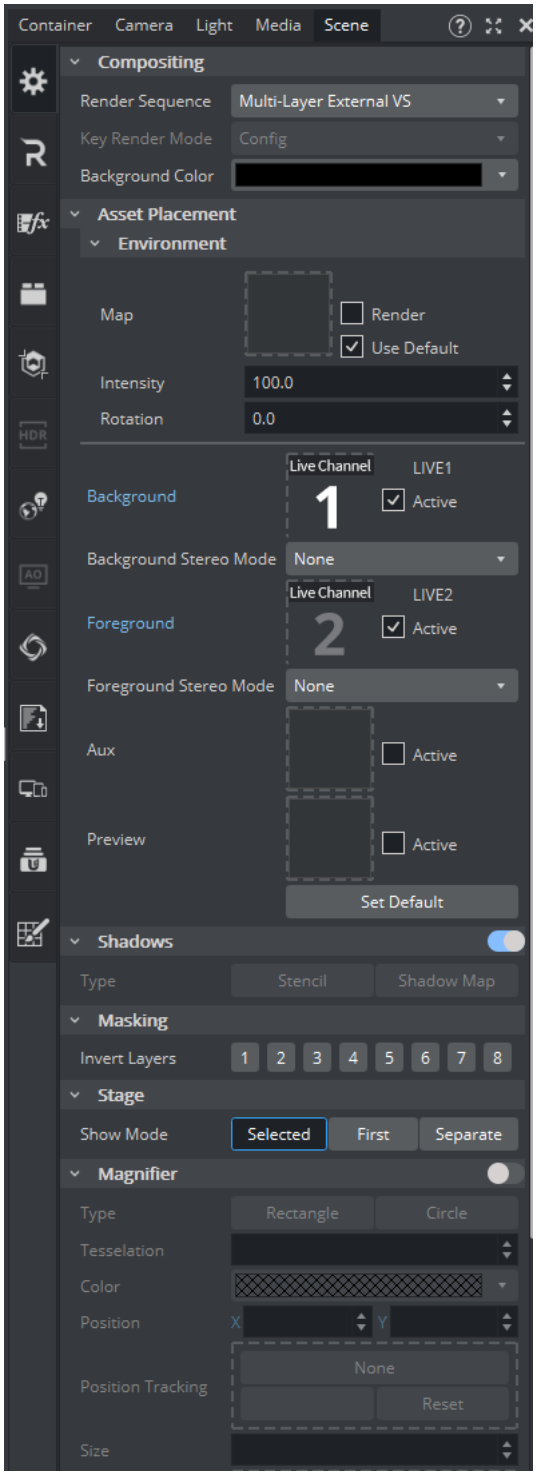
4.2 Scene Tab

Note: The Scene Tab is independent of the AI Keyer setup. The rest of this guide shows an example configuration of the Scene Tab.

For this example, make sure to set the **Scene > Render Engine** to *Viz Engine Renderer*.



Set the **Scene > Render Sequence** to *Multi-Layer External VS*. Drag the original source (Live Channel 1 in this example) to the background and the SHM - SMURF coming from the AI Keyer as the foreground (Live Channel 2 in this example).



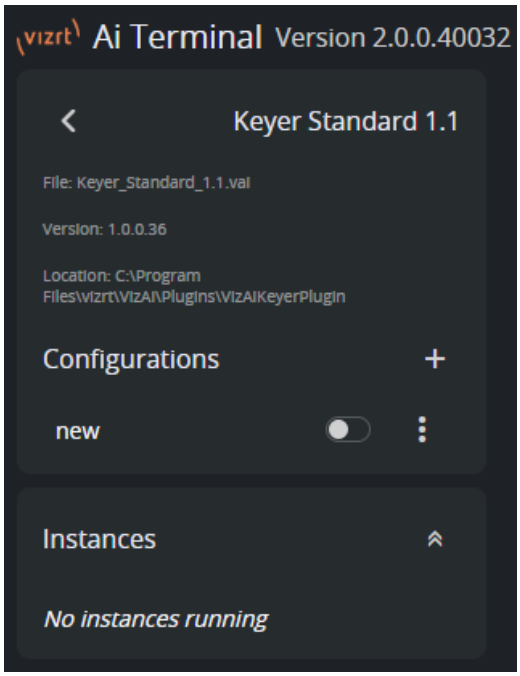
You can now start designing the scene with a correctly keyed talent:

5 Tool Configuration

5.1 Create a Viz AI Runner Instance

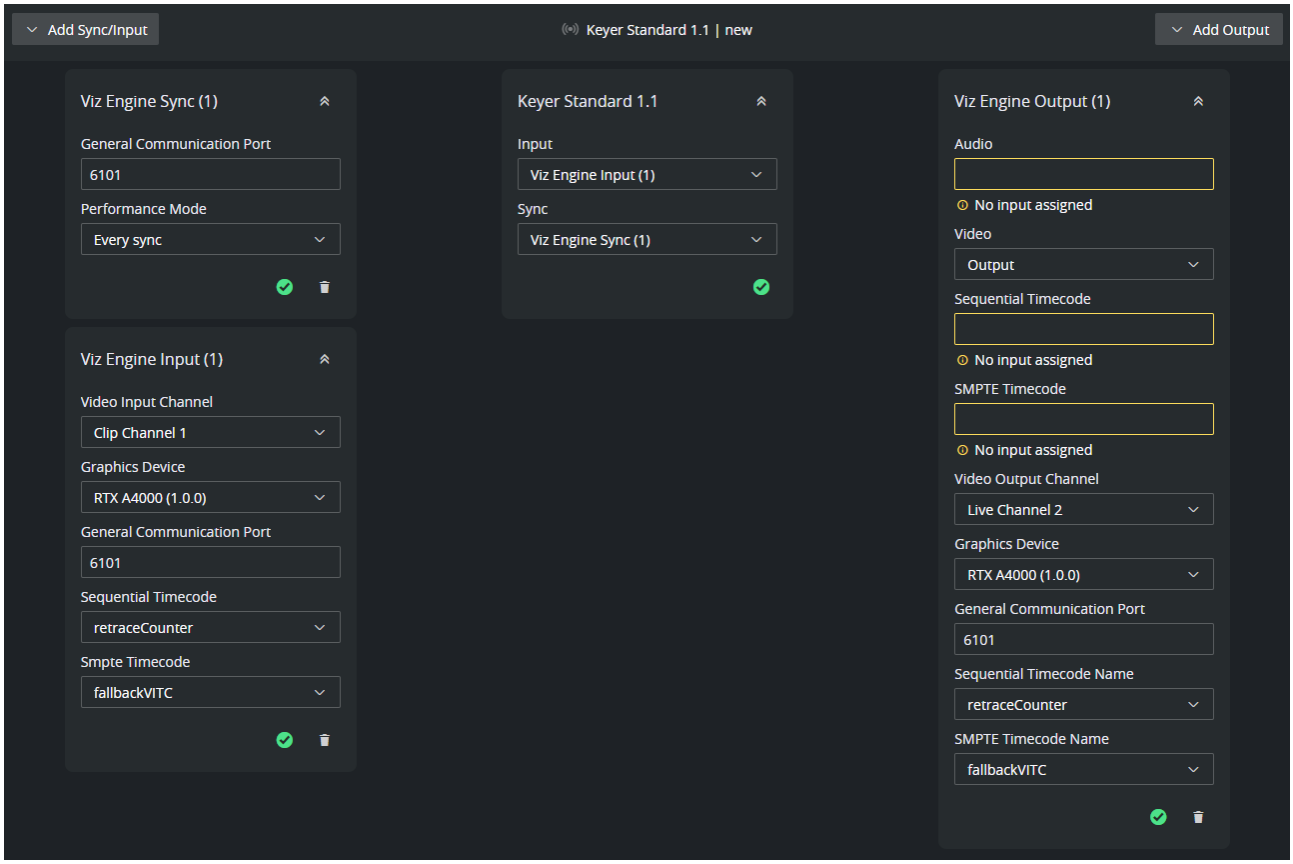
After installation, *Keyer Standard 1.0*, *Keyer Standard 1.1*, and *Keyer Studio 1.1* models are automatically discovered by the AI Terminal. Choose one, then:

- Create a new configuration with the + button.
- Launch with an existing configuration using the power on/off switch.



Once the instance is powered on, a Viz AI Runner console becomes visible and its configuration can be opened under *Instances*.

5.2 Input / Output Configuration



5.2.1 Viz Engine Sync Input

- Specify the same value for **General Communication Port** as configured earlier in Viz Engine Configuration.
- Set the **Performance Mode** to *Every sync*. If performance problems appear later when starting the tool, this mode can be changed to improve performance.

5.2.2 Viz Engine Input

- Set the **Video Input Channel** to the configured live video source.
  - The dropdown menu is automatically populated and refreshed every 60 seconds.
  - In this guide, we assume that Live Channel 1 is the live video input source, as shown in the Viz Engine Configuration section.
- Set **Graphics device** to the device the Viz Engine is running on.
- Specify the same value for **General Communication Port** as configured earlier in Viz Engine Configuration.
- Choose the **Sequential Timecode** and **SMPTE Timecode** to use (for example, *retraceCounter* and *fallbackVITC*, respectively).

5.2.3 Keyer Standard / Studio

- Set the **Input** to the respective **Viz Engine Input**.
- Set the **Sync** to **Viz Engine Sync Input**.

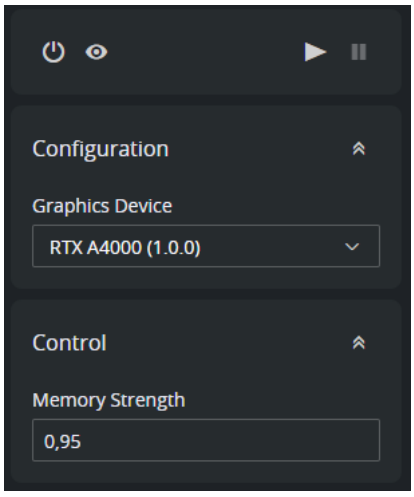
5.2.4 Viz Engine Output

- Set the **Video** to *Output.*
- Set the **Video Output Channel** to the configured SHM - SMURF source.
  - The dropdown menu is automatically populated and refreshed every 60 seconds.
  - In this guide, we assume that Live Channel 2 is the SHM - SMURF source, as shown in the Viz Engine Configuration section.
- Set **Graphics Device** to the device the Viz Engine is running on.
- Specify the same value for **General Communication Port** as configured earlier in Viz Engine Configuration.
- Choose the **Sequential Timecode Name** and **SMPTE Timecode Name** to use, as set in the **Viz Engine Input** panel (for example, *retraceCounter* and *fallbackVITC*, respectively)

Note: Entries for **Audio**, **Sequential Timecode** and **SMPTE Timecode** can be left blank.

Note: In rare cases, the Tool Configuration interface stops refreshing the **Video Input Channel** or **Video Output Channel** dropdowns**,** and an empty list is shown. In this case, to force a refresh, please manually close the Tool console and start the configuration again. Parameters already entered remain.

5.3 Tool Configuration



5.3.1 Configuration

- Select the **Graphics Device** that is used by the AI Keyer tool.

> **Note:** For the *Keyer Studio 1.1* model it is highly recommended to run the AI Keyer on another graphics device than the Viz Engine is running on.

5.3.2 Control

Choose the desired **Memory Strength** value. This value affects how much information is retained into the model "memory" context and is applied immediately.

- A value of `0.0` is equivalent to running the AI Keyer with no memory ("one-shot" prediction).
- A value of `1.0` is equivalent to running the AI Keyer with the maximum memory context.

> **Information:** Setting a value for the **Memory Strength** depends on the specific application. A memory strength of 0 helps removing *ghosting* artifacts that can otherwise appear in the case of fast moving objects or hard-cut scenarios. On the other hand, it can introduce flickering in the produced key, due to lack of temporal context. Similarly, a memory strength of 1.0 helps reducing flickering over temporal sequences, risking introducing ghosting artefacts in fast-moving scenarios. Start with a **Memory Strength** of 0.95 and only adjust it if needed.

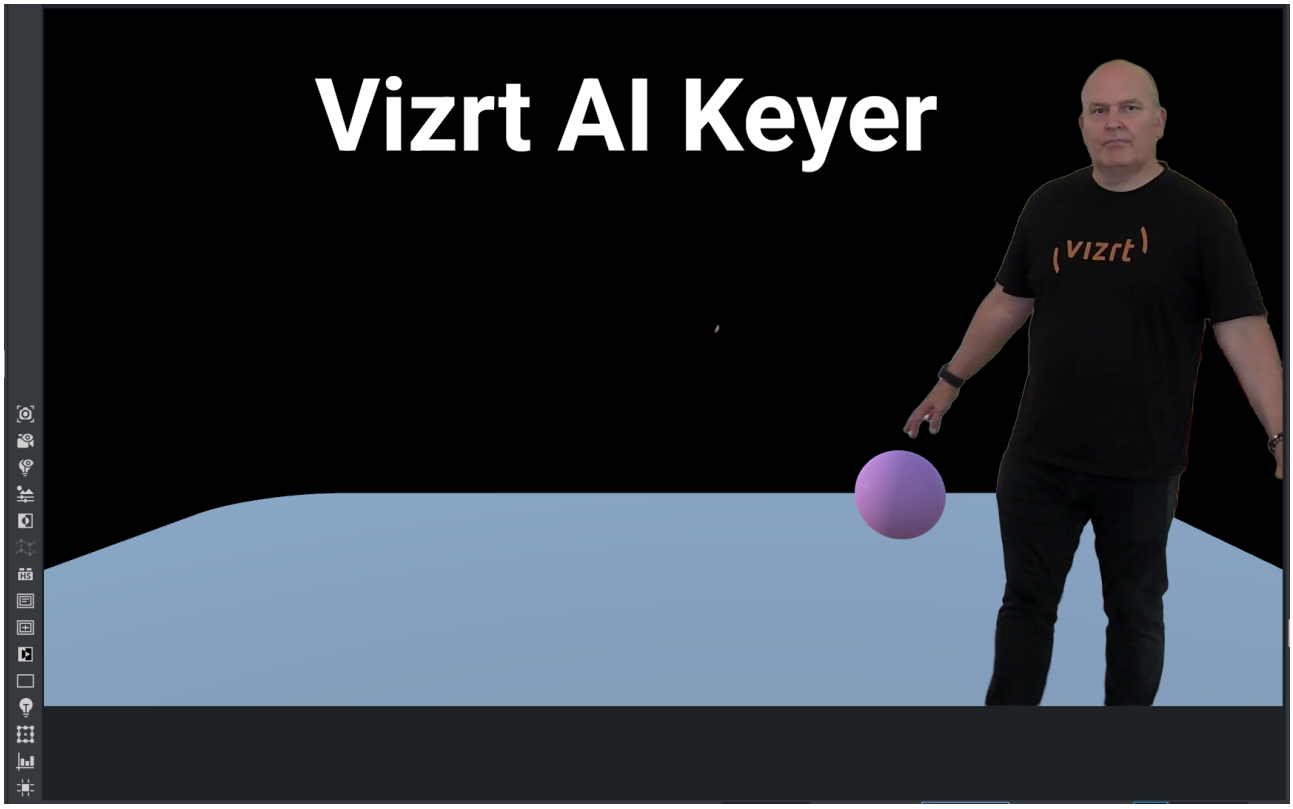
Refer to Models for the individual model's additional parameters.

> **Note:** While not running, the Tool periodically contacts Viz Engine to retrieve information. While this operation happens sporadically, to avoid consuming resources make sure to close all Viz AI Runner consoles when not in use.

5.4 Verify Configuration

Start the tool using the play button ▶ in the top right corner of the page and verify that no errors are shown. Check the *AI Terminal User Guide* for proposed actions in case of errors.

If all applications are setup as instructed, a scene designed as in Viz Engine Scene Design displays a correctly keyed talent.



This can be verified by clicking the **Key ON/OFF** button in the Engine Scene Editor window to preview the incoming key: