Service Host

Version 1.0

Created on Nov 15, 2017

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.

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

11/15/2017

Table of Contents

1 Getting Started

Service Host is a component that abstracts the Windows Service layer. The Service Host operates with Service Host plugins that implement the services and functionality, while the Service Host itself manages Windows Service related operations, such as logging, on behalf of all plugins.

An example is the **Control Mode** plugin. By default, it registers as a Windows service named `ServiceHost.control` upon successful installation. **Control Mode** provides a REST interface and comes with a web-based user interface, and is used to operate the Service Host plugin instances and installed Viz Engine instances.

The fastest way to get started with Service Host is:

1. Install Service Host, see Install / Uninstall .
2. Launch the web GUI of the Control Mode. After each launch of The Control Mode, a shortcut to the web GUI endpoint is updated under `%ProgramData%/vizrt/ServiceHost/ServiceHost.control.` Open this shortcut with your favorite browser. A description of this web GUI is found under Control Mode .

2 Channel Recorder

This is the Administrator's Guide for **Channel Recorder**, containing details regarding the installation, configuration and operation of Channel Recorder.

Channel Recorder is a service that creates video clips, using an SDI or IP stream as source. It can be controlled using an arbitrary set of commands, or by remote applications utilizing the Multiport Video Computer Protocol (MVCP).

2.1 Installation and Configuration

This section describes the installation requirements and procedure required to use Channel Recorder. It contains information on the following topics:

2.1.1 Installing, Registering and Removing Channel Recorder

Channel Recorder runs as a plugin of Service Host. Channel Recorder cannot operate without Service Host, for this reason the installation of Channel Recorder consists of the following parts:

1. Running the **Service Host Setup Wizard**.
2. Manually registering one or more Channel Recorder instance(s) using the Service Host web interface.

Before installing Service Host, please make sure to acquire the latest installer from Vizrt's FTP server: ftp://download.vizrt.com/

To install Service Host

Refer to the Service Host documentation.

To Register Channel Recorder instance

Refer to the Service Host documentation.

After successfully registering your Channel Recorder instance, the following dialog window appears:
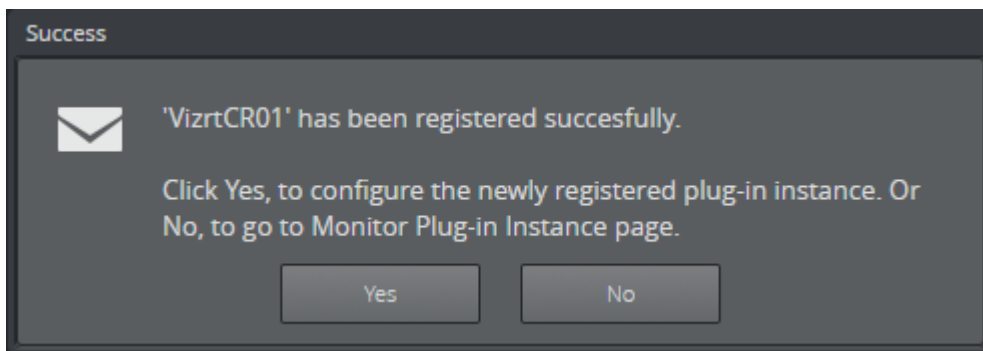
This is to inform you that the registration of the instance is successful. You are prompted to choose between configuring the instance now or later. See Channel Recorder Configuration (see page 8) for further details.

Upgrading or Repairing an Existing Installation

Use the Service Host Setup Wizard to upgrade or repair an existing installation of Channel Recorder.

When upgrading or repairing, Service Host remembers the registered instances and which states they were before the upgrade or repair occurs. After a successful upgrade or repair operation, any previously running services stopped by the Setup Wizard is restarted.

Removing Channel Recorder

Use the Service Host web interface to remove Channel Recorder. For more information, refer to the Service Host documentation.

2.1.2 Channel Recorder Configuration

You can configure Channel Recorder at any time after successfully registering an instance. For any configuration changes to take effect, you must restart the instance.

Every plugin of Service Host has its own configuration page. For more information, refer to the Service Host documentation.

Every plugin configuration page consist of two main sections:

- Plugin Configuration Section (see page 9)
- Service Host Section (see page 13)

The Plugin Configuration Section is unique for each plugin.

From the Service Host Section, you can set specific command arguments to the plugin. This is useful mainly for debugging purposes.
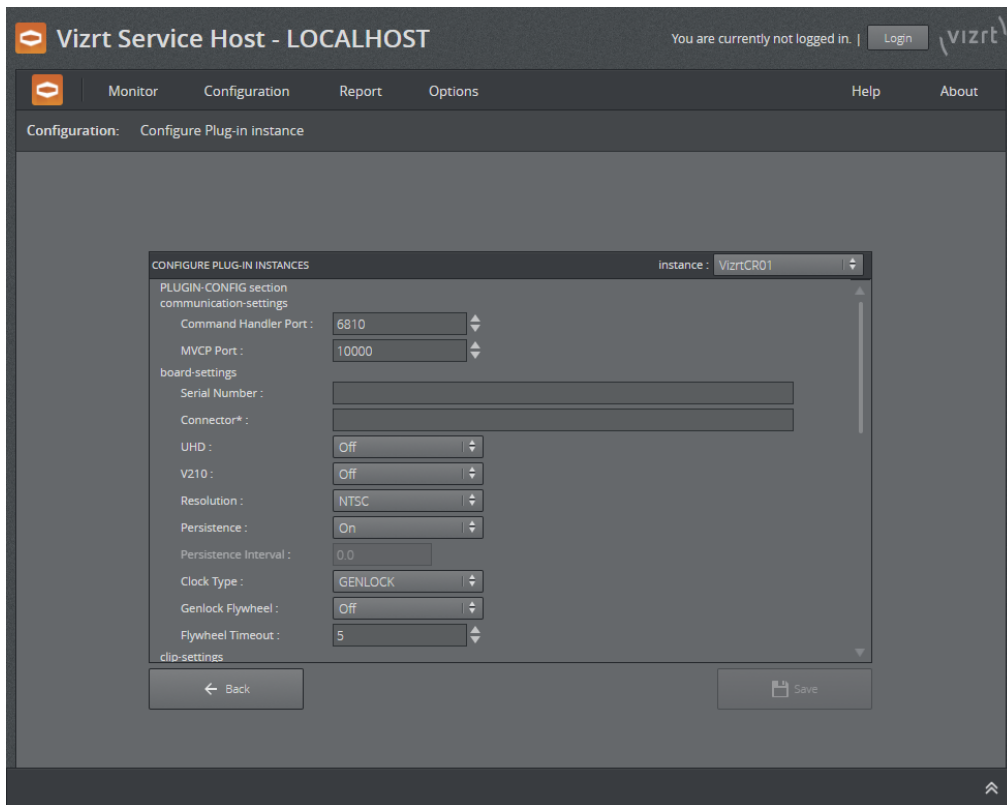
Plugin Configuration Section

The Plugin Configuration Section consists of 3 parts:

- **Communication Settings** (changes to these settings only take effect by restarting the Channel Recorder instance).
    - **Command Handler Port**: This is the port used to communicate with Channel Recorder via **Viz Send**. The default value is `6810`.
    - **MVCP Port**: This is the port used to communicate with Channel Recorder using MVCP. Some Vizrt products, such as Viz Dart, use this protocol to communicate with Channel Recorder. The default value is `10000`.

- **Board Settings** (changes to these settings only take effect by restarting the Channel Recorder instance).
    - **Serial Number**: Select the board with the specified serial number. If no serial number is specified, the first detected board is selected. By default, no value is specified.

> ⚠️ **Tip:** The serial number on Matrox video boards can be found in the hardware tab of **Matrox X.info** and it usually starts with an A.

- **Connector**: This will select the connector that will be used for recording live input. This is a required value. The default value is *empty*.
- **UHD**: Enable detection of UHDTV signals. When set to 0n, the Channel Recorder scans the signal resolution on the four corresponding input connectors. If four 3G signals are detected, they are interpreted as one UHDTV signal. When set to Off, the four connectors are treated as separate 3G signals. The default value is Off.
- **V210**: Use the 10-bit surface format V210. This is needed to record XAVC. It also increases performance when for example recording ProRes. This surface format is not supported on the Matrox X.mio2+. The default value is Off.
- **Resolution**: Set the default resolution. The possible values are: NTSC, PAL, 720p50, 720p60M, 1080i25, 1080p60, 1080i30M, 1080i30, 1080p50, 1080p60M and 1080p60. The default value is NTSC.
- **Persistence**: Choose if a backup of the scheduled recording is needed. This is only useful for scheduled recordings. The default value is Off.
- **Persistence Interval**: Define, in seconds, the interval in which the backup is written to disk. The default value is 0.
- **Clock Type**: Define what type of clock is used for recording. The possible values are: GENLOCK and INPUT. The default value is GENLOCK.

> ⚠️ **Tip:** INPUT clock type is recommended when the GENLOCK signal is unstable. INPUT clock also allows recording when no input signal is detected.

- **Genlock Flywheel**: If Clock Type is GENLOCK, also use the flywheel in case the genlock signal is lost.
- **Flywheel Timeout**: Set the timeout of the genlock flywheel in seconds. This defines the time until the genlock switches to free run, as well as the maximum time the flywheel can use to resynchronize. The default value is 5.0.

- **Clip Settings** (some of these settings can be changed when Channel Recorder is running. Refer to Channel Recorder Control Commands).
    - **File Extensions**: Enable or disable automatically adding a file extension to the file name. If this feature is turned off, the client application has full control over the file name. The default value is `Off`.
    - **Trigger Threshold**: If a timed command misses the execution time, but is still within the trigger threshold, it will be executed (late). Outside of this window, the command is ignored until the next time the timecode is received. The value can either be a number of frames or a timecode-based relative value. The default value is `5`.
    - **Bitrate**: Set the bitrate for the video encoding. The value can either be applied as bps (bits per second) or as mbps (megabits per second). Not all codecs allow changes to the bitrate. In such cases, this value is ignored. The default value is `0`.
    - **Stop At End**: Set the default value at the end of recording. The default value is `Off`.
    - **TDIR**: Enable or disable Time Delayed Instant Replay. The default is `On`.
    - **Duration**: Set the default duration of the recording. The default value is `0`.
    - **TDIR Interval**: Set the interval of file header updates in TDIR recordings. The value is in seconds and fractions of seconds, meaning both 11.1 and 11.2 are considered valid values. Minimum allowed value is 10.0, which is interpreted by Channel Recorder as every frame. The maximum value is 60.0. The default value is `10.0`.
    - **Update Modification Time**: Update the modification time of the recorded clip regardless of TDIR setting value. The default value is `On`.
    - **Audio Channels**: Set the number of audio channels to record. How many channels are actually recorded depends on the codec and the input signal. The default value is `2`.
    - **Writers**: Set the number of writers to initialize. The default value is `2`.

    > ⚠ **Tip:** More writers increase memory consumption however it will prevent failed recording when the scheduled recordings are very close to each other in the order of 5 to 10 seconds.

    - **Clip Root**: Set the default folder for the recordings. The default value is `V:/`.
    - **Disk Access Size**: Set the size of data blocks written to the disk in byte. Postfixes like KiB, Kb, k, etc., are allowed, but must not be separated from the value with a blank space. The default value is `8MiB` (2*4194304 bytes). The minimum value is `32KiB` (32768 bytes).

    > ⚠ **Tip:** KiB and k multiplies the value by 1024. kb multiplies the value by 1000. The same works with m for mega and g for giga.

    - **Ring Buffer**: Set the size of the capture ring buffer. The default value is *60*.
    - **Timeout**: Set the timeout for the capture operation in milliseconds. If the recorder reports timeout errors, increasing the timeout could help. The value can be in frames or in timecode format: 00:00:00:00. The default value is `0`.
    - **Codec**: Set the codec type of the recorded file. The possible values are: DvCam, DvCPro, Dv50, IFrame, XDCam, AVCIntra50, AVCIntra100, ProRes. The default value is `XDCam`.
    - **Container**: AVCINTRAMXF, AVI, DVCPROMXF, MOV, MXF, XAVCMXF, XDCAMMXF.
    - **Priority**: Set the process priority class. The values correspond to the Windows process priority. The default value is `Normal`.

- **Timecode Log Interval**: Specify the interval at which the current timecode is logged. The value can either be a number of frames or a timecode-based relative value. The default value is `0`, which means that every full second is logged.
- **Timecode Source**: Specify the timecode source. The default value is `VITC`.

- **Output Settings** (these settings can be changed when Channel Recorder is running. Refer to Channel Recorder Control Commands).
    - **Output**: Specify the output mode. The possible values are: **VIDEO**, **CODER**, **NONE**. The default value is `NONE`.
    - **Output Arguments**: Arguments for the selected output mode. For **CODER** it is `[shared memory name]`, `[proxy hostname]`, `[proxy port]` and for **VIDEO** it is `[connector name]`.

> ⚠ **Tip:** Only one type of output can be configured at startup using the web interface. However, after startup, it is still possible to configure another output via Viz Send.

Service Host Section

One setting can be set in this section:

- **Arguments**: Specify the arguments that are going to be passed to Channel Recorder.

An example of a string that can be used for enabling log level debug is: `-v -l debug`

See Also

- Channel Recorder Control Commands
- Channel Recorder Startup Options
- Example Configuration File

2.1.3 Example Configuration File

You can manually change the configuration, without using the web interface as described in Channel Recorder Configuration .

The configuration file is located in `%programdata%\vizrt\ServiceHost\`.

The name of the configuration file is the **Service Name** that has been used during the registration of the instance.

> ⊘ Manual manipulation of the configuration file is prone to errors. For this reason you are strongly advised not to do it. Whenever possible, use the web interface provided by Service Host.

```xml
<payload xmlns="http://www.vizrt.com/types" model="model.xml">
        <field name="plugin-config">
                <field name="communication-settings">
                        <field name="command-handler-port">
                                <value>6810</value>
                        </field>
                        <field name="mvcp-port">
                                <value>10000</value>
                        </field>
                </field>
                <field name="board-settings">
                        <field name="serial-number" />
                        <field name="connector" />
                        <field name="uhd">
                                <value>0</value>
                        </field>
                        <field name="v210">
                                <value>0</value>
                        </field>
                        <field name="resolution">
                                <value>NTSC</value>
                        </field>
                        <field name="persistence">
```

```xml
                <value>1</value>
        </field>
        <field name="persistence-interval">
                <value>0.0</value>
        </field>
        <field name="clock-type">
                <value>GENLOCK</value>
        </field>
        <field name="genlock-flywheel">
                <value>0</value>
        </field>
        <field name="flywheel-timeout">
                <value>5</value>
        </field>
</field>
<field name="clip-settings">
        <field name="file-extension">
                <value>1</value>
        </field>
        <field name="trigger-threshold">
                <value>5</value>
        </field>
        <field name="bitrate">
                <value>0</value>
        </field>
        <field name="stop-at-end">
                <value>0</value>
        </field>
        <field name="tdir">
                <value>1</value>
        </field>
        <field name="duration">
                <value>0</value>
        </field>
        <field name="tdir-interval">
                <value>10.0</value>
        </field>
        <field name="update-mt-enable">
                <value>0</value>
        </field>
        <field name="audio-channels">
                <value>2</value>
        </field>
        <field name="writers">
                <value>2</value>
        </field>
        <field name="clip-root">
                <value>V:/</value>
        </field>
```

```xml
                              <field name="disk-access-size">
                                      <value>8MiB</value>
                              </field>
                              <field name="ring-buffer">
                                      <value>60</value>
                              </field>
                              <field name="timeout">
                                      <value>0</value>
                              </field>
                              <field name="codec">
                                      <value>XDCam</value>
                              </field>
                              <field name="container">
                                      <value>MXF</value>
                              </field>
                              <field name="priority">
                                      <value>NORMAL</value>
                              </field>
                              <field name="timecode-log-interval">
                                      <value>0</value>
                              </field>
                              <field name="timecode-source">
                                      <value>VITC</value>
                              </field>
                      </field>
                      <field name="output-settings">
                              <field name="output">
                                      <value>VideoOut</value>
                              </field>
                              <field name="output-arguments">
                                      <value>A</value>
                              </field>
                      </field>
              </field>
              <field name="service-host">
                      <field name="plugin">
                              <field name="plugin-name">
                                      <value>ChannelRecorder</value>
                              </field>
                              <field name="plugin-arguments" />
                      </field>
              </field>
      </field>
</payload>
```

2.1.4 Integration with Other Services

Channel Recorder can be integrated with other services.

Viz Dart

The Viz Dart video acquisition tool can be configured to acquire assets using Channel Recorder. Integration of Channel Recorder with Viz Dart is based on the MLT Video Control Protocol, or MVCP. MVCP is by default activated on port **10000**. To change the port, refer to Channel Recorder Configuration (see page 8).

For further information on how to configure Viz Dart, or how to operate it on a Video Disk Recorder, refer to the Viz Dart Documentation.

Coder

Coder is the next generation transcoder that can be used as a standalone component with Viz Engine, or in a MAM environment using Viz One. To setup the Channel Recorder to output to Coder, issue the following command:

```
OUTPUT START Coder [shared memory name] [proxy hostname] [proxy port]
```

Alternatively, the output can be started automatically during startup. Refer to Output Settings in Channel Recorder Configuration (see page 8).

[proxy hostname] and [proxy port] are only used by Viz Coder Recording Proxy and are not necessary. This application automatically creates a job for Coder and transcodes using the setting specified in an XML file bundled with the application. If Viz Coder Recording Proxy is not used [proxy hostname] and [proxy port] can be left out.

To be able to use Coder with Channel Recorder in a new machine follow these steps:

1. Install Viz Coder
2. Go to C:\Program Files (x86)\vizrt\Coder and run **coder_slave.exe** in the cmd with Administrator privileges:

   ```
   register_slave.exe http://[IP]:[Port]
   ```

   Where [IP] IP address where Coder is installed (if in doubt use the localhost IP) and [Port] is the port that Coder is using (if in doubt use 8081 which is the default value of Coder port).
3. Go to C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Vizrt\Coder and run **run_benchmark.exe** by double clicking on it.

   > ⊘ Do not run run_benchmark.exe from C:\Program Files (x86)\vizrt\Coder since this will not work

   The port used by Coder can be checked in C:\ProgramData\vizrt\Coder\vizrt-coder-master.log
4. Go to Coder web interface http://[IP]:[Port]/ui/war/index.html[1] (e.g. http://localhost:8081/ui/war/index.html) and check if the slave worker is there by clicking the Workers tab.
5. Create a new live job with the following setting:

---

1 http://localhost:8081/ui/war/index.html

a. Input URL: shm://@[IP]:[SHMName] where [IP] is the IP address where Channel Recorder is located and [SHMName] is the shared memory name given in the configuration without the *Global* string.

> ⚠ To use Coder the shared memory name given in the configuration of Channel Recorder must be preceded by *Global* this is because Coder always expects the shared memory to be located in the Global namespace.

b. Output URL: udp://@[IP]:[Port] where [IP] and [Port] is the IP address and port where the stream is going to be sent
c. Profile URL: mpegts_mpeg2_720p

> ⚠ The settings above are just example and can be changed, however it is recommended to test with these values first before experimenting with other values

To test that the above steps are done correctly, open VLC and go to Media → Open Network Stream and use the address specified in Output URL when Channel Recorder is recording and using the appropriate configuration. Example of configuration for Coder:

```
<field name="output-settings">
        <field name="output">
                <value>Coder</value>
        </field>
        <field name="output-arguments">
                <value>Global\viz_shm_cr_01</value>
        </field>
</field>
```

Please note that `Global\` is used for the name of the shared memory. For the Input URL only `viz_shm_cr_01` should be given.

For further information on the configuration and operation of Coder, refer to the Media Service documentation.

See Also

- Media Service documentation
- Viz One documentation for Viz Dart
- Channel Recorder Configuration
- References and Specifications

2.1.5 Hardware and Software Requirements

Channel Recorder runs on the following hardware:

- HP Z840
- Dell R7910 (2U) Rack Server

The Viz Engine Administrator's Guide provides a detailed description of these machines.

Channel Recorder requires one of the following Video Cards:

- Matrox X.mio3 IP
- Matrox X.mio3
- Matrox DSX LE4
- Matrox X.mio2+

Channel Recorder requires Matrox DSX.utils version 9.8.1.22400.

See Also

- Viz Engine Administrator's Guide

2.1.6 Removing Channel Recorder

Channel Recorder can be removed by using the web interface provided by Service Host. For more information please refer to the Service Host documentation.

See Also

- Channel Recorder Startup Options

2.1.7 Upgrading from 1.0

The biggest difference between version 1.0 and 1.1 of Channel Recorder is that Service Host is now the front end controller. Starting with version 1.1, Channel Recorder is considered a plugin of Service Host. This means that the Channel Recorder is managed by Service Host through the web interface that it provides. With Service Host you can:

- Register a new instance
- Remove an existing instance
- Start an existing instance
- Stop an existing instance
- Provide an **initial configuration** to an instance.

For more information, refer to the Service Host documentation.

Configuration file

For users upgrading from Channel Recorder 1.0, the only required step is the translation of the configuration into the new format. Previously, the configuration was provided via a simple text file located in `%programdata%/vizrt/Channel Recorder`.

This file might contain the following:

```
COMMAND_HANDLER PORT SET 6810
MAIN BOARD SELECT A523502
MVCP PORT 10000
CONFIG SET RINGBUFFERSIZE 60
MAIN CLOCKTYPE GENLOCK
MAIN CONNECTOR SELECT A
```

```
CONFIG SET FILEEXTENSION OFF
CONFIG SET PRIORITY REALTIME
CONFIG SET CLIP_ROOT D:/
CONFIG SET TIMEOUT 5000
CONFIG SET AUDIOCHANNELS 4
CONFIG SET CODEC XDCamHD
CONFIG SET CONTAINER XDCAMMXF
CONFIG SET TIMECODE TIME_OF_DAY
CONFIG SET TDIRENABLE FALSE
CONFIG SET VBI OFF
```

In Channel Recorder 1.1, the configuration is provided via an xml file located in `%programdata%/vizrt/ServiceHost`. The configuration file name is the **Service Name** used when registering the Channel Recorder instance. This xml file can be edited directly, however, it is highly recommended that you use the Service Host web interface.

All configuration settings that was in 1.0 are still present with the same parameters and similar names.

You must use the Service Host web interface and configure Channel Recorder appropriately to achieve the same behavior as in the old version. Configuration is described in Channel Recorder Configuration.

Do not forget to restart the instance after any changes to the configuration.

Log file

As with the configuration file, the log file is now located in `%programdata%/vizrt/ServiceHost` instead of `%programdata%/vizrt/Channel Recorder.`

The old file used to configure the logging mechanism is gone. The only option in the new version is to change the logging level. Refer to the Service Host Section in Channel Recorder Configuration.

2.2 Operation and Troubleshooting

You can use the application **Viz Send**, which is included in **Viz Artist** installation, to communicate with Channel Recorder. Connect the tool to the port specified in the **Channel Recorder** configuration file.

2.2.1 Channel Recorder Initialization

You can start Channel Recorder with the Service Host web interface.

When Channel Recorder is started, the following will happen:

1. User specified configuration runs in a specific order:
    a. If no serial number is specified, Channel Recorder selects the first board detected by the system.
    b. If no connector is selected, Channel Recorder fails to execute.

2. Depending on the selected clock type, one of the following can happen:
    a. If GENLOCK is selected, Channel Recorder queries for the genlock signal. Channel Recorder compares the genlock frequency to the signal frequency. The signal frequencies need to be from the same frequency family, for example, `25i` and `50p` frequencies will match, but `29.97i` and `50p` will not.

    b. If INPUT is selected, Channel Recorder does not query for the genlock signal.

> ⚠️ **Note: INPUT** clock type is only available for Matrox X.mio3 IP, Matrox X.mio3 and Matrox DSX LE4.

If everything works as expected, the input channel is set up and Channel Recorder starts capturing from it. As long as no clip is recorded, the captured frames are thrown away. When a recording starts, these frames are written to the file. This is necessary to be able to provide:

1. Instant recording
2. Scheduling of recordings

2.2.2 Workflow Example

1. Start the service from the Service Host web interface. For more information, refer to the Service Host documentation.
2. Check the log in the Service Host web interface to make sure that everything has started correctly. You can also check the log file located at `%programdata%vizrt/ServiceHost/[service name]`.
3. You should not see any errors or warning messages. If there is any error or warning messages, please try to amend it and restart the service using the web interface provided by Service Host. For more information, refer to the Service Host documentation. The following is an example of a successful initial run by Channel Recorder:

```
[2017-10-05 13:54:52.140 +02:00][10700][info]: Startup
'ChannelRecorder::ChannelRecorder::Connect' as 'VizrtCR01'
[2017-10-05 13:54:52.147 +02:00][21528][info]: [VizrtCR01] starting up. thread-id: 21528
[2017-10-05 13:54:52.147 +02:00][21528][info]: [VizrtCR01] Applying settings from
configuration file!
[2017-10-05 13:54:52.147 +02:00][21528][info]: [VizrtCR01] Received: -1 COMMAND_HANDLER
PORT SET 6810
[2017-10-05 13:54:52.154 +02:00][21528][info]: [VizrtCR01] setting COMMAND_HANDLER
PORT=6810
[2017-10-05 13:54:52.154 +02:00][21528][info]: [VizrtCR01] Received: -1 MVCP PORT 6000
[2017-10-05 13:54:52.154 +02:00][21528][info]: [VizrtCR01] setting MVCP PORT=6810
[2017-10-05 13:54:52.155 +02:00][21528][info]: [VizrtCR01] Received: -1 MAIN BOARD SELECT
A520889
[2017-10-05 13:54:53.136 +02:00][21528][info]: [VizrtCR01] Received: -1 CONFIG SET
PERSISTENCE 0
[2017-10-05 13:54:53.136 +02:00][21528][info]: [VizrtCR01] setting PERSISTENCE=false
[2017-10-05 13:54:53.136 +02:00][21528][info]: [VizrtCR01] Received: -1 MAIN CLOCKTYPE
INPUT
[2017-10-05 13:54:53.136 +02:00][21528][info]: [VizrtCR01] Received: -1 MAIN CONNECTOR
SELECT A
```

```
[2017-10-05 13:54:53.347 +02:00][21528][info]: [VizrtCR01::CaptureChannelSystemInA] Could
not find connector in existing topologies. Creating a new one...
[2017-10-05 13:54:53.347 +02:00][23412][info]:
[VizrtCR01::CaptureChannelSystemInA::Scheduler::FileManager] starting up
[2017-10-05 13:54:53.356 +02:00][21528][info]: [VizrtCR01::CaptureChannelSystemInA] Detect:
Selected resolution is 1080i25
[2017-10-05 13:54:53.363 +02:00][21528][info]: [VizrtCR01] Capture from channel InA on
board XMIO3/X2/550 (serial: A520889)
[2017-10-05 13:54:53.363 +02:00][20704][info]: [VizrtCR01::CaptureChannelSystemInA]
starting up
[2017-10-05 13:54:53.363 +02:00][21528][info]: [VizrtCR01] Received: -1 CONFIG SET
TDIRENABLE 0
[2017-10-05 13:54:53.363 +02:00][21528][info]: [VizrtCR01::CaptureChannelSystemInA] setting
TDIRENABLE=false
[2017-10-05 13:54:53.364 +02:00][21528][info]: [VizrtCR01] Received: -1 CONFIG SET
TDIRINTERVAL 15.0
[2017-10-05 13:54:53.364 +02:00][21528][info]: [VizrtCR01::CaptureChannelSystemInA] setting
TDIRINTERVAL=15
[2017-10-05 13:54:53.364 +02:00][21528][info]: [VizrtCR01] Received: -1 CONFIG SET
AUDIOCHANNELS 4
[2017-10-05 13:54:53.364 +02:00][21528][info]: [VizrtCR01::CaptureChannelSystemInA] setting
AUDIOCHANNELS=4
[2017-10-05 13:54:53.365 +02:00][21528][info]: [VizrtCR01] Received: -1 CONFIG SET
CLIP_ROOT C:/
[2017-10-05 13:54:53.365 +02:00][21528][info]: [VizrtCR01::CaptureChannelSystemInA] setting
CLIP_ROOT=C:/
[2017-10-05 13:54:53.365 +02:00][21528][info]: [VizrtCR01] Received: -1 CONFIG SET
DISKACCESSSIZE 8MiB
[2017-10-05 13:54:53.366 +02:00][21528][info]: [VizrtCR01::CaptureChannelSystemInA] setting
DISKACCESSSIZE=8388608
[2017-10-05 13:54:53.366 +02:00][21640][info]:
[VizrtCR01::CaptureChannelSystemInA::ClipOutChannel0] starting up
[2017-10-05 13:54:53.367 +02:00][26060][info]:
[VizrtCR01::CaptureChannelSystemInA::ClipOutChannel1] starting up
[2017-10-05 13:54:53.367 +02:00][21528][info]: [VizrtCR01] Received: -1 CONFIG SET
RINGBUFFERSIZE 100
[2017-10-05 13:54:53.367 +02:00][21528][info]: [VizrtCR01] setting RINGBUFFERSIZE=100
[2017-10-05 13:54:53.369 +02:00][21528][info]: [VizrtCR01] Received: -1 CONFIG SET TIMEOUT
100
[2017-10-05 13:54:53.369 +02:00][21528][info]: [VizrtCR01::CaptureChannelSystemInA] setting
TIMEOUT=100
[2017-10-05 13:54:53.370 +02:00][21528][info]: [VizrtCR01] Received: -1 CONFIG SET CODEC
AVCIntra100
[2017-10-05 13:54:53.370 +02:00][21528][info]: [VizrtCR01::CaptureChannelSystemInA] setting
CODEC=AVCIntra100
[2017-10-05 13:54:53.370 +02:00][21528][info]: [VizrtCR01] Received: -1 CONFIG SET
CONTAINER AVCINTRAMXF
[2017-10-05 13:54:53.370 +02:00][21528][info]: [VizrtCR01::CaptureChannelSystemInA] setting
CONTAINER=AVCINTRAMXF
```

```
[2017-10-05 13:54:53.371 +02:00][21528][info]: [VizrtCR01] Received: -1 CONFIG SET PRIORITY
REALTIME
[2017-10-05 13:54:53.371 +02:00][21528][info]: [VizrtCR01] Received: -1 CONFIG SET TIMECODE
TIME_OF_DAY
[2017-10-05 13:54:53.371 +02:00][21528][info]: [VizrtCR01::CaptureChannelSystemInA] setting
TIMECODE=TIME_OF_DAY
[2017-10-05 13:54:53.372 +02:00][21528][info]: [VizrtCR01] Received: -1 OUTPUT START
VideoOut A
[2017-10-05 13:54:53.377 +02:00][24120][info]: [VideoOut] starting up
```

4. The most common errors are not specifying a connector, or specifying a connector that is already in use or non-existing.

When Channel Recorder is successfully running, you can control it using Viz Send, or by using Viz Dart through the MVCP protocol. Refer to Channel Recorder Control Commands for available commands.

> ⚠ **Note:** Telnet can also be used as a way to control Channel Recorder through the MVCP protocol.

2.2.3 Change of Input Resolution

Starting with version 1.1, the Channel Recorder handles input resolution changes during runtime. When Channel Recorder detects a change in the input resolution, it resets the channel to use the new resolution, without the need to restart the service. This requires that you select **INPUT** as **Clock Type** during initial configuration.

2.2.4 Recording with no Input Signal

Starting with version 1.1, the Channel Recorder can record without an input signal attached. When a signal is attached during recording, Channel Recorder will seamlessly use the signal. This means that the recorder file will have a black signal, plus the new connected signal. This requires that the configured resolution is the same as the connected signal resolution, and that you select the **INPUT Clock Type** during initial configuration.

2.2.5 Drop Frame Timecode

The user should be mindful when using commands to operate Channel Recorder than involves inputting timecode in drop frame resolutions. Such commands require the input of a valid timecode. However, when a timecode is not recognized as valid, Channel Recorder will try to fix it by rounding down the timecode to a a valid one. For example:

RECORD SET schedule.mxf IN=15:00:00:00 OUT=16:00:00:00

Both timecodes are not valid when operating in drop frame resolution. The command and timecode will be automatically changed to:

RECORD SET schedule.mxf IN=15:00:00:04 OUT=16:00:00:04

See Also

- Channel Recorder Control Commands
- Channel Recorder Startup Options

- Channel Recorder Configuratio (see page 8)n

2.2.6 Troubleshooting Channel Recorder

This section contains some common troubleshooting tips:

- ChannelRecorder and Viz Engine (see page 23)
- The Service Reports Timeout Errors (see page 23)
- Channel Recorder fails to initialize (see page 23)
- Recording UHD with XAVC produces specified compression format is not supported error (see page 23)

ChannelRecorder and Viz Engine

As both Channel Recorder and Viz Engine are using a Matrox topology, make sure Viz Engine is being started before Channelrecorder if you need to use a shared input. If Engine is started afterwards, the acquisition of the selected input connector will fail.

The Service Reports Timeout Errors

If Channel Recorder reports timeout errors, try increasing the timeout value by sending the command `CONFIG SET TIMEOUT [timeout]`, or by configuring the timeout value using the web interface provided by Service Host.

Channel Recorder fails to initialize

Channel Recorder will fail to initialize if GENLOCK is selected as clock type and no valid synchronization signal or input signal is connected to the Matrox video board. Make sure that the Matrox board is connected to an input and sync source.

Recording UHD with XAVC produces specified compression format is not supported error

UHD recording with XAVC requires a Matrox M264 board to be installed in the system. If the system does not have such a board installed, Channel Recorder will report that the specified compression format is not supported.

2.2.7 Logging

The log file for Channel Recorder is located at `%programdata%/vizrt/ServiceHost`. The log file name reflects the **Service Name** the Channel Recorder instance is registered as in Service Host.

For troubleshooting, you can enable a higher level of logging for more information. For more information, refer to the Service Host Section in Channel Recorder Configuration.

2.2.8 Crash Recording

Channel Recorder can be used to instantly record any input signal into a clip file. The delay between executing the command and the first field written to the file is usually less then 2 frames.

The following is an example of how to invoke a crash recording with MVCP commands:

```
UADD CS_ENCODER1 * SHAR Capture
LOAD U1 "MVCP_Crashrecording" IN CRTE NOEX
/SEQA SET U1 MED vtr.media.input.trigger.vitc.in² "0"
CUER U1
REC U1

USTA U1
STOP U1
UNLD U1
UCLS U1
BYE
```

With regular Channel Recorder commands:

```
CONFIG SET CONTAINER XDCAMMXF
CONFIG SET CODEC XDCam
RECORD CLIP Crash.mxf
RECORD START
RECORD STOP
```

2.2.9 Loop Recording

You can split recordings into various chunks. These can either be specified via timecode format, or with fields being written.

The following is an example of how to run a Loop Recording using regular Channel Recorder commands:

```
RECORD LOOP CHUNK=00:01:00:00
RECORD LOOP PREFIX=Loopy
RECORD LOOP START
```

This will generate chunks of 1 minute until you stop the recording.

Possible options for loop recordings are:

- LENGTH Length of the loop in frames or timecode. At least this amount of frames will reside on the disk. The minimum length value is 00:02:00:00 or the equivalent number of fields / frames.
- CHUNK The size of one chunk in frames or timecode. The chunk size influences the loop length. The minimum chunk value is 00:00:30:00 or the equivalent number of fields / frames.

---

2 http://vtr.media.input.trigger.vitc.in

- `DISKSIZE` The size of the loop is defined by the size of the clips on the disk. The oldest clips are deleted until the disk size drops below the specified value.
- `DISKFREE` The size of the loop is defined by the space left on the disk. If it drops below the value, the oldest clips are deleted until at least the specified amount is free again.

The following is an example on how `LENGTH` and `CHUNK` parameters are related:

```
RECORD LOOP CHUNK=00:00:30:00
RECORD LOOP LENGTH=00:04:00:00
RECORD LOOP PREFIX=Loopy
RECORD LOOP START
```

This generates a total of 9 files over 4 minutes. 1 of the files will always be 0 bytes, as this is the next file the Channel Recorder will write to. Another file will have a changing size, this is the file Channel Recorder is currently writing to. The other 7 files have already been written. When Channel Recorder reaches 4 minutes of recording, it will start removing the first file it has written.

2.2.10 Scheduled Recording

Channel Recorder has its own scheduler to plan recordings in future.
The command to record an input signal starting at 12:00 and running for 1 hour would be the following:

> *RECORD SET scheduled.mxf IN=12:00:00:00 OUT=13:00:00:00*

If a scheduled recording is being interrupted by a Loop or Crash recording, the following logic applys:

- A recording is being issued while a scheduled recording is running: **The scheduled recording will be interrupted and stopped.**
- During loop recording someone decides to schedule a recording in the future: **The recording will be scheduled and run when/if the loop recording stops before the specified time, otherwise it will fail.**

2.3 References and Specifications

This section details references and specifications for the Channel Recorder.

This section contains the following topics:

2.3.1 Channel Recorder Startup Options

It is possible to reroute some commands to Channel Recorder during initialization. Refer to the Service Host Section in Channel Recorder Configuration. The only usage for this is to activate a higher level of logging.

2.3.2 Channel Recorder Control Commands

The application **Viz Send**, included in **Viz Artist** installation, can be used to communicate with Channel Recorder. Connect the tool to the port specified in the configuration file. The following commands are implemented in the service:

- ABOUT (see page 26)
- MAIN (see page 26)
- OUTPUT (see page 27)
- RECORD (see page 28)
- CONFIG (see page 33)
- CONFIG SET (see page 33)
- COMMUNICATION (see page 38)
- EXIT (see page 39)

ABOUT

Command	Description
ABOUT GET	Print license information of all libraries used in this software.

MAIN

Some of the following commands must be called during initialization. These commands are striked through. To configure these commands, use the configuration web interface and the restart the service. Refer to Channel Recorder Configuration for more information.

Command	Description
MAIN VERSION	Return application version.
GET VERSION	Same as MAIN VERSION. Return application version.
MAIN HOSTNAME	List all available boards.
MAIN SVCNAME	Select a board. The board is identified by the serial number. If the board with the given number is not found, the first board will be selected.

MAIN BOARD LIST	List all available boards.
MAIN BOARD SELECT [serial number]	Select a board. The board is identified by the serial number. If the board with the given number is not found, the first board will be selected.
MAIN BOARD GET	Return the selected boards.
~~MAIN CONNECTOR SELECT~~ ~~[connector number]~~ ~~([address] [port])~~	Select a connector. Connectors are labeled consecutively from A to H. The `[address]` and `[port]` parameters are required for IP boards only, and defines which IP address and UDP port the Channel Recorder will listen to for the input stream.
MAIN CONNECTOR GET	Return the selected connector.
~~MAIN CLOCKTYPE~~ ~~[type]~~	Select the type of clock. [type] can be one of: `INPUT`**:** Use the clock on the input signal. **With this mode the record can start without any input or genlock signal being connected.** If there is no input signal black will be recorded. `GENLOCK`**:** Use the genlock signal. This is the default type and what was being used in 1.0.
MAIN CRASH [type]	Crash the service. [type] can either be omitted or be one of: `MAIN`**:** Crash the service (same as when omitted). `RECORD`**:** Crash the recording thread.

OUTPUT

One output target can be initialized during startup. During execution more than one output can be enabled. For instance you can have VideoOut A and Coder enabled. Please refer to Channel Recorder Configuration to initialize an output during startup.

Command	Description
OUTPUT GET	List the active targets.

OUTPUT LIST	List available targets.

OUTPUT START [target]	Start an output handler for the specified [target]. Several output handlers can be started in parallel, but only one is allowed for each target. Valid targets are: VideoOut: This target requires one additional parameter: [connector]. The [connector] parameter specifies the video output connector of the Matrox board.
	
	⚠ **Example:**
	    OUTPUT START VideoOut A
	
	Coder: This target requires three additional parameters, [SHM name], [address] and [port].
	[SHM name] is the name of the shared memory area the fields will be written to. The name cannot contain whitespace characters.
	[address] denotes the IP address or host name of the proxy service, and [port] is used to specify which port the service is listening to.
	
	⚠ **Example:**
	    OUTPUT START Coder SHMCoder1 localhost 12345

OUTPUT STOP [target]	Stop the output handler for the specified target.

RECORD

Crash recording / one-time scheduling

These are specific commands used only for crash and one-time scheduling of recording. These command are available from version 1.0. For scheduling it is recommended to use RECORD SET introduced from version 1.1.

Command	Description
RECORD CLIP [clip name]	Set the clip name and initialize the recorder. This command does not start recording (see RECORD START).

RECORD START [duration] [start time] [end mode]	Start or continue recording. The [duration], [start time], and [end mode] parameters are optional. However, the parameters are interdependent as follows: If the parameter [start time] is provided, [duration] is also required. If the parameter [end mode] is provided, [duration] is also required. The format for both is **Timecode** (see Data Types (see page 39)). [duration] and [start time] can be zero timecode, which will then be ignored. [end mode] can either be STOP, which finalizes and closes the clip, or PAUSE (default behavior), where the clip stays open and can be used for further recording.
RECORD PAUSE	Pause recording.
RECORD STOP [end time]	Stop recording and flush the recorder. A new clip needs to be set afterwards (see RECORD CLIP). The end time is optional and specifies the timecode when the recording should end.

> ⚠ **Example crash recording:**
>
> RECORD CLIP crashRecording.mxf
>
> RECORD START
>
> RECORD STOP
>
> **Example one-time scheduling recording:**
>
> RECORD CLIP crashRecording.mxf
>
> RECORD START 600 15:00:00:00

Loop recording

These are specific commands used only for loop recording. It allows setup, start or stop loop recording. These are introduced from version 1.1.

Command	Description
RECORD LOOP [key=value] … [key=value] [operation]	Start, stop or configure the settings for loop recording. Any number of properties can be applied in the form of key-value-pairs. If no property is supplied, and the entry does not yet exist, it is added with the default values. Available keys are: • `PREFIX Prefix to the filename.  It will be appended with the timestamp of the start time.` • `POSTFIX Postfix to the filename.  Will be appended after the timestamp.` • `LENGTH Length of the loop in frames or timecode.  At least this amount of frames will reside on the disk. The minimum length value is 00:02:00:00 or the equivalent number of fields / frames.` • `CHUNK The size of one chunk in frames or timecode.  The chunk size influences the loop length. The minimum chunk value is 00:00:30:00 or the equivalent number of fields / frames.` • `DISKSIZE The size of the loop is defined by the size of the clips on the disk.  The oldest clips are deleted until the disk size drops below the specified value.` • `DISKFREE The size of the loop is defined by the space left on the disk.  If it drops below the value, the oldest clips are deleted until at least the specified amount is free again.` [operation] can be START or STOP. When no [operation] is specified then only the settings are set for the specified key-value-pairs. It is possible to specify [key=value] [operation] at the same time but only when [operation] is START. ⚠️ **Example:** RECORD LOOP LENGTH=00:05:00:00 CHUNK=00:00:30:00 PREFIX=Loopy START RECORD LOOP STOP

Scheduled recording

These are specific commands used only for scheduled recording. It allows setup, start or stop scheduled recording.

Command	Description
RECORD SET [name] [key= value] … [key= value]	Add a new clip to the timeline, or change a property of an entry defined by `[name]`. Any number of properties can be applied in the form of key-value-pairs. If no property is supplied, and the entry does not yet exist, it is added with the default values. Available keys are: • `CODEC Set the codec type of the recorded file. By setting the codec type, default values for bitrate and audio will also be set.` • `CONTAINER Set the container type of the recorded file. With the container type, a default codec is also set. By setting the container type, a valid recording can be started.` • `BITRATE Set the bitrate for the video encoding. The value can either be applied as bits per second or as Megabits per second. Not all codecs allow changes to the bitrate. In such cases, BITRATE will be ignored.` • `IN Set the timecode when the recording should start.` • `OUT Set the timecode when the recording should stop.` • `DURATION Set the default value for the recording duration. The initial value is 0.` • `VBR Enable or disable VBI` • `TDIRENABLE Set the default behavior of Time Delayed Instant Replay (TDIR).` • `AUDIOCHANNELS Set the number of audio channels to record. How many channels are actually recorded depends on the codec and the input signal.` • `STARTTC Defines which start timecode to use for the recording.` For more information regarding these settings possible values and default values check the section CONFIG SET below.
RECORD REMOVE [name] … [name]	Remove the entry `[name]`. Multiple `[name]` arguments can be provided.

RECORD GET [name] | [key] ... [name] | [key]	List all entries or show the properties of an entry. If no parameter is applied, a list of all scheduled clips is returned. Any number of [name] and [key] can be applied. The values of all keys will be returned for all values. If no [key] is provided, all values of the applied keys will be returned and vice versa. This means that if you call RECORD GET with clip names only, the command returns all properties of these clips. When called with properties only, it will return this property for all clips. A special [key] is TIMELINE, which returns all entries in the timeline.

Common record commands

Some of these commands are only used for loop and crash recording, while others are used for all three modes.

Command	Description
RECORD DURATION	If no duration parameter is applied to the command, it **returns the timecode relative to the start timecode**. If a duration parameter is applied, the duration of an ongoing recording is changed. The duration parameter can be specified as either a number of frames or a timecode (see Data Types (see page 39)). In the case of loop and schedule recording **the value used to set** using this command is ignored.
RECORD RESOLUTION GET	Returns the resolution the Channel Recorder is running at.

RECORD STATE	Returns the state the Record Channel service is in. By default state is idle. State is only invalid when the clip was configured incorrectly (e.g. container / codec combination is wrong). ⚠ **State flow:**
	
	```Idle -- Start() --> Recording -- Stop() --> Paused```
	`` ^
		|```
		``` `-- Flush() ----------+----------------------´```

RECORD START TC [starttc]	Defines which start timecode to use for the recording. If the command is not called before the recording starts, the current timecode is written to the clip. If no value is given, the command returns the current value. This is only used for loop and schedule recording.

CONFIG

Command	Description
CONFIG GET	Get the current settings.
CONFIG LIST [argument]	List all available variables. The optional `[argument]` flag provides information specific to the applied argument. Known arguments are: • `CONTAINER` - displays all known container types. • `CODEC` - displays all known codec types. • `DUMP` - displays the current dump file setting. • `PRIORITY` - displays the current process priority setting.
CONFIG SET [variable] [value]	Set the `[variable]` to `[value]`. To get all available variables, call `CONFIG LIST`. Refer also to variables and values for `CONFIG SET` below.

CONFIG SET

Variabled and values for CONFIG SET command. The CONFIG SET command is used to set the configuration globally, this means that all the recording mode will use these settings by default unless specified otherwise. Only schedule recording can specify different settings for each schedule by specifiying them via RECORD SET command.

Variable	Value
AUDIOCHANNELS [channels]	Set the number of audio channels to record. How many channels are actually recorded depends on the codec and the input signal.
BITRATE [rate]	Set the bitrate for the video encoding. The value can either be applied as bits per second or as Megabits per second. Not all codecs allow changes to the bitrate. In such cases, `BITRATE` will be ignored.

`CLIP_RO`
`OT`
`[path]`
	Set the root folder for the captured files. If the filename in command `RECORD CLIP` does not contain a absolute path name, the `CLIP_ROOT` is prepended.

`CODEC`
`[codec`
`type]`
	Sets the codec type of the recorded file. By setting the codec type, default values for bitrate and audio will also be set. These values can be changed (see below). Not all codec types are available with all container types.
	Available codec types are:
	
	• `Default`
	• `DvCam`
	• `DvCPro`
	• `Dv50`
	• `IFrame`
	• `XDCam (See note regarding bitrates specification)`
	• `XDCamEX (See note regarding bitrates specification)`
	• `XDCamHD (See note regarding bitrates specification)`
	• `AVCIntra50`
	• `AVCIntra100ProRes`
	
	⚠ **Note 1:** To utilize the TDIR capabilities of the **ProRes** codec with a **.mov** container in Viz Engine, you must use the `.Ref`-file
	
	Note 2: When container is XDCAMMXF the default codec will be XDCam which is HD422. The default bitrate for XDCam, XDCamEX and XDCamHD are 50, 35 and 35 or 25 repectively. XDCamEX will output 1920x1080 clip at 35 Mbps and XDCamHD will output 1440x1080 clip at 35 Mbps (VBR) or 25 Mbps (CBR) depending on the bitrate chosen.

`CONTAIN ER [contai ner type]`	Sets the container type of the recorded file. With the container type, a default codec is also set. By setting the container type, a valid recording can be started.
	Known container types are:
	`Default`
	`AVCINTRAMXF` - see note regarding audio and timecode support
	`AVI`
	`DVCPROMXF` - see note regarding audio and timecode support
	`MOV`
	`MXF` - see note regarding audio and timecode support
	`XAVCMXF` - Requires a Matrox M264 card in order to work. Also, see note regarding audio and timecode support.
	`XDCAMMXF`- see note regarding audio and timecode support
	
	⚠️ **Note:** The `MXF` container type uses the **OP-Atom** format and does not include audio or timecode information. However, the `AVCINTRAMXF`, `DVCPROMXF`, `XAVCMXF` and `XDCAMMXF` container types all use the **OP1a** format, which features audio and timecode support. Refer to the Supported Codecs (see page 41) for further details.

`DISKACC ESSSIZE [size]`	Set the size of data blocks written to the disk in byte. Postfixes like KiB, Kb, k, etc., are allowed, but must not be separated from the value with a blank space (see example). The default value is **4 MiB** (4194304 bytes). The minimum value is **32 KiB** (32768 bytes). `KiB` and `k` multiplies the value by `1024`.`kb` multiplies the value by `1000`. The same works with `m` for mega and `g` for giga. ⚠️ **Example:** `CONFIG SET DISKACCESSSIZE 1024KiB`- sets the data block size to 1048576 bytes (one mebibyte).
`DUMP [NORMAL | PRIVATE | FULL]`	Set the dump file content to the specified value: `NORMAL`: Include just the information necessary to capture stack traces for all existing threads in a process. `PRIVATE`: Includes the contents of every readable and writable private memory page. `FULL`: Include all accessible memory in the process.
`DURATIO N [frames | timecod e]`	Set the default value for the recording duration. The initial value is `0`. When applying a duration to the `RECORD START` command, the default value is ignored but not changed. Refer to Data Types (see page 39) for information on time code formatting.

`FILEEXT ENSION [boolea n]`	Enables or disables automatically adding a file extension to the file name. If this feature is turned off, the client application has full control over the file name. The default value is `ON`.Please refer to Data Types (see page 39) for details on accepted boolean values.
`PERSINTE RVAL [seconds ]`	Set the interval, at which the data is written to the disk. The value is in seconds and fractions of seconds, meaning both `0.1` and `1.0` are considered valid values. A value of `0` means that every change is written. The default value is `0`.
`PERSISTE NCE [bool]`	Turn on persistence for operational data like the config and the timeline. The default value is `ON`.

`PRIORIT Y [priori ty]`	Sets the process priority class. The values correspond to the Windows process priority. Valid values for `[priority]` are:
	`Default`
	`IDLE`
	`BELOW_NORMAL`
	`NORMAL = Default`
	`ABOVE_NORMAL`
	`HIGH`
	`REALTIME`

`RINGBUF FERSIZE [size]`	Set the size of the capture ring buffer. The default value is `60`.
`RESOLUT ION [re solutio n]`	Set the default resolution. Format of `[resolution]` is: `WxHs@F, WxHs@FM, Hs@F, Hs@FM, NTSC, PAL` With `W` = Width, `H` = Height, `s` = Scanmode, `F` = Framerate, and `M` = Drop Frame Flag. Possible values for `s` are: • `i` or `I`: Interlaced • `p` or `P`: Progressive • `psf` or `PSF`: Progressive segmented `M` is optional, and as an alternative, you can use framerate with two decimal points (e.g. `30M` = 29.97). Examples: `NTSC, PAL, 1280x720p@50, 1920X1080P@50, 1280x720p@60M, 1280x720p@59.97, 720p50, 720P60M, 1080P50`. This flag must be set before selecting a connector using MAIN CONNECTOR SELECT (see page 0). The default value is `NTSC`.

`STOPATE`
`ND`
`[boolea`
`n]`
	Set the default behavior at the end of the recording.
	The default setting is `NO/FALSE/OFF`.
	Refer to Data Types (see page 39) for details on accepted boolean values.

`TCLOGINT`
`ERVAL`
`[frames|`
`timecode`
`]`
	Specify the interval at which the current timecode is logged. The value can either be a number of frames or a timecode-based relative value. The default value is 0, which means that every full second is logged.

`TDIRENA`
`BLE`
`[boolea`
`n]`
	Set the default behavior of Time Delayed Instant Replay (TDIR).
	The default setting is `ON`.

`TDIRINT`
`ERVAL`
`[time]`
	Set the interval of file header updates in TDIR recordings. The value is in seconds and fractions of seconds, meaning both `0.1` and `1.0` are considered valid values. Minimum allowed value is `0.001`, which is interpreted by Channel Recorder as every frame. A typical value would be `10.0`, the default value is `3.0`.
	
	⚠ **IMPORTANT!** In order to secure proper operation with Viz Engine, this value must **not** exceed `10.0`.

`TIMECOD`
`E`
`[source`
`]`
	Specify the timecode source. Valid values for `[source]` are:
	`Default`
	`ZERO_BASED`: The timecode written to the file starts at `00:00:00:00`.
	`VITC` = `Default`: Write the ATC/VITC from the input signal to the file.
	`LTC`: Write the ATC/LTC from the input signal to the file.
	`TIME_OF_DAY`: Write the current system time to the file. In order to get a proper time from the system, the system time must be synchronized using a time server.
	
	⚠ **Note:** From version 1.1 Channel Recorder is able to read SMPTE 12M-1 VITC timecode from PAL and NTSC signals

`TIMEOUT`
`[timeco`
`de]`
	Set the timeout for the capture operation in milliseconds. If the recorder reports timeout errors, increasing the timeout could help.
	Refer to Data Types (see page 39) for information on time code formatting.

`TRIGGERT HRESHOLD [frames| timecode ]`	If a timed command misses the execution time, but is still within the trigger threshold, it will be executed (late). Outside of this window, the command is ignored until the next time the timecode is received. The value can either be a number of frames or a timecode-based relative value. The default value is 5 frames.
`UHD [bool]`	Enable detection of UHDTV signals. When set to `ON`, the Channel Recorder scans the signal resolution on the four corresponding input connectors. If four 3G signals are detected, they are interpreted as one UHDTV signal. When set to `OFF`, the four connectors are treated as separate 3G signals. This flag must be set before selecting a connector using `MAIN CONNECTOR SELECT` (see page 0). The default value is `OFF`.
`VBI [bool]`	Turn off VBI recording. The default value is `ON`. This flag must be set before selecting a connector using `MAIN CONNECTOR SELECT`.
`V210 [bool]`	Use the 10-bit surface format V210. This is needed to record XAVC. It also increases performance when for example recording ProRes. This surface format is not supported on the Matrox X.mio2+. This flag must be set before selecting a connector using `MAIN CONNECTOR SELECT` (see page 0). The default value is `OFF`.
`WRITERS [number ]`	Set the number of writers to initialize. More writers increases memory consumption. This flag must be set before selecting a connector using `MAIN CONNECTOR SELECT` (see page 0). The default value is 2.

COMMUNICATION

Command	Description
`COMMAND_HANDLER PORT SET [port]`	Set the port of the command interface. A port can only be set once.
`COMMAND_HANDLER DUMP`	Prints this list of available commands.
`MVCP PORT [port]`	Set the port for the MVCP communication. This command is required to initialize MVCP.

EXIT

Command	Description
exit, EXIT	Stop all channels, clean up the hardware and stop the service.

2.3.3 Data Types

Booleans

Booleans are a data type with only two possible values; true or false. The Channel Recorder accepts YES, NO, TRUE, FALSE, ON, or OFF as boolean values, where:

- YES, TRUE and ON are positive values switching a flag **on**, and
- NO, FALSE and OFF are negative values switching a flag **off**.

Timecodes

Timecodes in Channel Recorder have the form <hh>:<mm>:<ss>:<ff>, where

- <hh> represents the hour in 24 hour format,
- <mm> is minutes,
- <ss> is seconds, and
- <ff> refers to the frame number, according to the frame rate.

> ⚠ **Example:** A zero timecode looks like this: 00:00:00:00.

2.3.4 Files and Directories

The Channel Recorder uses various files and directories while running.

Description	Type	Location
Installation Directory	Directory:	`<viz install folder>\ServiceHost`
ProgramData Directory	Directory:	`%ProgramData%\vizrt\ServiceHost`
Configuration	File:	`<instance_name>.conf`
	Path:	`%ProgramData%\vizrt\ServiceHost\<instance_name>.conf`

2.3.5 Supported Multiport Video Computer Protocol (MVCP) Commands

The Channel Recorder supports the following sets of Multiport Video Computer Protocol (MVCP) commands:

Supported Global MVCP Commands

Command	Function
BYE	Closes the current connection.
MON	Event monitoring mode.
PLS	Returns the supported media ports.
UADD	Create a new unit.
ULS	Returns the existing VST units.

Supported Unit MVCP Commands

Command	Function
CUER	Cue recording of the unit's currently loaded clip.
LOAD	Load a clip into a unit.
SET	Set controls for the unit: **vtr.media.clip.format**: Allowed values: `[movie/stream/mxf]` **vtr.media.video.input.compression.type** `[mpg2]` **vtr.media.mpeg.bit_rate**: Set the recording bitrate. **vtr.media.mpeg.imx**: Enable IMX recording. **vtr.media.audio.input.channels**: Set the number of audio channels to record. **vtr.media.audio.input.sample.width**: Set audio bits per sample. **vtr.media.video.standard**: The command is understood, however, as the video standard is determined by the input signal, it does not get changed. **vtr.media.input.trigger.duration.out**: Set the duration of the recording. **vtr.media.clip.start.mode**: Allowed values: `[time-of-day][vitc]`
STOP	Stop playback or recording.

UCLS	Close a unit.
UNLD	Unloads the unit's currently loaded clip.
USTA	Returns the status of a unit.

See Also

- The Video Server Toolkit Developer's Guide[3] on ftp.sgi.com.

2.3.6 Supported Codecs

The Channel Recorder supports the following containers and codecs.

- PAL Codecs (see page 42)
  - MPEG-IFrame (see page 42)
  - DVCAM (see page 42)
  - DVCPRO (see page 43)
  - XDCAM (see page 44)
  - Uncompressed (see page 44)

- NTSC Codecs (see page 45)
  - MPEG IFrame (see page 45)
  - DVCAM (see page 45)
  - DVCPRO (see page 46)
  - XDCAM (see page 48)
  - Uncompressed (see page 48)

- 720p50 and 720p59.94 (60M) Codecs (see page 48)
  - MPEG-IFrame (see page 48)
  - DVCPRO HD (see page 49)
  - XDCAM HD (see page 50)
  - AVC-Intra (see page 50)
  - ProRes (see page 51)
  - Uncompressed (see page 51)

- 1080i25 and 1080i29.97 (30M) Codecs (see page 52)
  - MPEG-IFrame (see page 52)
  - DVCPRO HD (see page 52)
  - XDCAM HD 422 (see page 53)
  - AVC-Intra (see page 53)
  - ProRes (see page 54)
  - Uncompressed (see page 54)

- 1080p50 and 1080p59.94 (60M) Codecs (see page 55)
  - AVC-Intra (see page 55)
  - MPEG-IFrame (see page 56)
  - ProRes (see page 56)
  - Uncompressed (see page 56)

---

3 ftp://ftp.sgi.com/sgi/vst/docs/VUG.pdf

-
  -

PAL Codecs

MPEG-IFrame

Container	Codec	Features	Audio
.avi	MPEG2-IFrame422 720x576	Alpha: No Bitrate: 10-50 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

DVCAM

Container	Codec	Features	Audio
.avi	DV/DVCAM 4:2:0 720x576	Alpha: No Bitrate: 25 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mov	DVCAM 4:2:0 720x576	Alpha: No Bitrate: 25 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: N/A	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mxf Panasonic P2(OP-Atom)	DVCAM 4:2:0 720x576	Alpha: No Bitrate: 25 Bit depth: 8 VBI: N/A Timecode: Yes TDIR: Yes	No Audio

DVCPRO

Container	Codec	Features	Audio
.avi	DVCPRO 4:1:1 720x576	Alpha: No Bitrate: 25 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz; - 4 ch: 24 in 32bit; 48kHz; - 8 ch: 24 in 32bit; 48kHz; - 16 ch: 24 in 32bit; 48kHz
.avi	DVCPRO 50 4:2:2 720x576	Alpha: No Bitrate: 50 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mov	DVCPRO 4:1:1 720x576	Alpha: No Bitrate: 25 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: N/A	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mov	DVCPRO 50 4:2:2 720x576	Alpha: No Bitrate: 50 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: N/A	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mxf (OP1a)	DVCPRO 4:1:1 720x576	Alpha: No Bitrate: 25 Bit depth: 8 VBI: N/A Timecode: Yes TDIR: Yes	PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mxf (OP1a)	DVCPRO 50 4:2:2 720x576	Alpha: No Bitrate: 50 Bit depth: 8 VBI: N/A Timecode: Yes TDIR: Yes	PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

.mxf Panasonic P2 (OP-Atom)	DVCPRO 4:1:1 720x576	Alpha: No Bitrate: 25 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio
.mxf Panasonic P2 (OP-Atom)	DVCPRO 50 4:2:2 720x576	Alpha: No Bitrate: 50 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio

XDCAM

Container	Codec	Features	Audio
.mxf XDCAM (OP1a)	D10 (IMX) 4:2:2 720x608	Alpha: No Bitrate: 50 Bit depth: 10 VBI: required Timecode: required TDIR: Yes	AES3 - 4ch: 24 in 32bit; 48kHz

Uncompressed

Container	Codec	Features	Audio
.avi	Uncompressed YUYV 4:2:2 720x576	Alpha: No Bitrate: 160 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

NTSC Codecs

MPEG IFrame

Container	Codec	Features	Audio
.avi	MPEG2-IFrame422 720x480	Alpha: No Bitrate: 10-50 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

DVCAM

Container	Codec	Features	Audio
.avi	DV/DVCAM 4:1:1 720x480	Alpha: No Bitrate: 25 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mov	DVCAM 4:1:1 720x480	Alpha: No Bitrate: 25 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: N/A	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mxf Panasonic P2 (OP-Atom)	DVCAM 4:1:1 720x480	Alpha: No Bitrate: 25 Bit depth: 8 VBI: N/A Timecode: Yes TDIR: N/A	No Audio

DVCPRO

Container	Codec	Features	Audio
.avi	DVCPRO 4:1:1 720x480	Alpha: No; Bitrate: 25; Bit depth: 8; VBI: N/A; Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz; - 4 ch: 24 in 32bit; 48kHz; - 8 ch: 24 in 32bit; 48kHz; - 16 ch: 24 in 32bit; 48kHz
.avi	DVCPRO 50 4:2:2 720x480	Alpha: No; Bitrate: 50; Bit depth: 8; VBI: N/A; Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mov	DVCPRO 4:1:1 720x480	Alpha: No Bitrate: 25 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: N/A	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mov	DVCPRO 50 4:2:2 720x480	Alpha: No Bitrate: 50 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: N/A	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

.mxf (OP1a)	DVCPRO 4:1:1 720x480	Alpha: No Bitrate: 25 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mxf (OP1a)	DVCPRO 50 4:2:2 720x480	Alpha: No Bitrate: 50 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mxf Panasonic P2 (OP-Atom)	DVCPRO 4:1:1 720x480	Alpha: No Bitrate: 25 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio
.mxf Panasonic P2 (OP-Atom)	DVCPRO 50 4:2:2 720x480	Alpha: No Bitrate: 50 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio

XDCAM

Container	Codec	Features	Audio
.mxf XDCAM (OP1a)	D10 (IMX) 4:2:2 720x512	Alpha: No Bitrate: 50 Bit depth: 10 VBI: Required Timecode: Required TDIR: Yes	AES3 - 4 ch: 24 in 32bit; 48kHz

Uncompressed

Container	Codec	Features	Audio
.avi	Uncompressed YUYV 4:2:2 720x486	Alpha: No Bitrate: 160 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

720p50 and 720p59.94 (60M) Codecs

MPEG-IFrame

Container	Codec	Features	Audio
.avi	MPEG2-IFrame422 1280x720	Alpha: No Bitrate: 50-300 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

DVCPRO HD

Container	Codec	Features	Audio
.avi	DVCPRO HD 1280x720	Alpha: No Bitrate: 100 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mov	DVCPRO HD 960x720	Alpha: No Bitrate: 100 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: N/A	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mxf (OP1a)	DVCPRO HD 960x720	Alpha: No Bitrate: 100 Bit depth: 8 VBI: N/A Timecode: Yes TDIR: Yes	PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mxf Panasonic P2 (OP-Atom)	DVCPRO HD 960x720	Alpha: No Bitrate: 100 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio

XDCAM HD

Container	Codec	Features	Audio
.mxf XDCAM (OP1a)	XDCAM HD422, IBP HD 4:2:2, 1280x720	Alpha: No Bitrate: 50 Bit depth: 8 VBI: N/A Timecode: Yes TDIR: N/A	PCM: - 8 ch: 24 in 32bit; 48kHz

AVC-Intra

Container	Codec	Features	Audio
.mxf (OP1a)	AVCIntra Class 50 960x720	Alpha: No Bitrate: 50 Bit depth: 8 VBI: N/A Timecode: Yes TDIR: Yes	PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mxf (OP1a)	AVCIntra Class 100 1280x720	Alpha: No Bitrate: 100 Bit depth: 8 VBI: N/A Timecode: Yes TDIR: Yes	PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mxf Panasonic P2 (OP-Atom)	AVCIntra Class 50 960x720	Alpha: No Bitrate: 50 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio

.mxf Panasonic P2 (OP-Atom)	AVCIntra Class 100 1280x720	Alpha: No Bitrate: 100 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio

ProRes

Container	Codec	Features	Audio
.mov	I-Frame 4:2:2 1280x720	Alpha: No Bitrate: 100 (LT), 147, 220 (HQ) Bit depth: 10 VBI: N/A Timecode: N/A TDIR: Yes, with `.Ref` file	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

Uncompressed

Container	Codec	Features	Audio
.avi	Uncompressed YUYV 4:2:2 1280x720	Alpha: No Bitrate: 700 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

1080i25 and 1080i29.97 (30M) Codecs

MPEG-IFrame

Container	Codec	Features	Audio
.avi	MPEG2-IFrame422 1920x1080	Alpha: No Bitrate: 50-300 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

DVCPRO HD

Container	Codec	Features	Audio
.avi	DVCPRO HD 1920x1080 *(1080i25)* 1260x1080 *(1080i29.97)*	Alpha: No Bitrate: 100 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mov	DVCPRO HD 1280x1080 *(1080i25)* 1260x1080 *(1080i29.97)*	Alpha: No Bitrate: 100 Bit depth: 10 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mxf Panasonic P2 (OP-Atom)	DVCPRO HD 1280x1080 *(1080i25)* 1260x1080 *(1080i29.97)*	Alpha: No Bitrate: 100 Bit depth: 10 VBI: N/A Timecode: N/A TDIR: Yes	No Audio

| .mxf (OP1a) | DVCPRO HD 1280x1080 *(1080i25)* 1260x1080 *(1080i29.97)* | Alpha: No
Bitrate: 100
Bit depth: 10
VBI: N/A
Timecode: Yes
TDIR: Yes | PCM:
- 4 ch: 24 in 32bit; 48kHz |

XDCAM HD 422

Container	Codec	Features	Audio
.mxf (OP1a)	XDCAM HD, IBP HD 4:2:2, Elementary, Program, Transport 1920x1080	Alpha: No Bitrate: 25 Bit depth: 8 VBI: N/A Timecode: Yes TDIR: Yes	PCM: - 8 ch: 16 in 16bit; 48kHz

AVC-Intra

Container	Codec	Features	Audio
.mxf (OP1a)	AVCIntra Class 50 1440x1080	Alpha: No Bitrate: 50 Bit depth: 8 VBI: N/A Timecode: Yes TDIR: Yes	PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mxf (OP1a)	AVCIntra Class 100 1920x1080	Alpha: No Bitrate: 100 Bit depth: 8 VBI: N/A Timecode: Yes TDIR: Yes	PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

.mxf Panasonic P2 (OP-Atom)	AVCIntra Class 50 1440x1080	Alpha: No Bitrate: 50 Bit depth: 10 VBI: N/A Timecode: N/A TDIR: Yes	No Audio
.mxf Panasonic P2 (OP-Atom)	AVCIntra Class 100 1920x1080	Alpha: No Bitrate: 100 Bit depth: 10 VBI: N/A Timecode: N/A TDIR: Yes	No Audio

ProRes

Container	Codec	Features	Audio
.mov	I-Frame 4:2:2 1920x1080	Alpha: No Bitrate: 100 (LT), 147, 220 (HQ) Bit depth: 10 VBI: N/A Timecode: N/A TDIR: Yes, with `.Ref` file	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

Uncompressed

Container	Codec	Features	Audio
.avi	Uncompressed YUYV 4:2:2 1920x1080	Alpha: No Bitrate: 800 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

1080p50 and 1080p59.94 (60M) Codecs

AVC-Intra

Container	Codec	Features	Audio
.mxf (OP1a)	AVCIntra Class 50 1440x1080	Alpha: No Bitrate: 100 Bit depth: 10 VBI: N/A Timecode: Yes TDIR: Yes	PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mxf (OP1a)	AVCIntra Class 100 1920x1080	Alpha: No Bitrate: 100 Bit depth: 10 VBI: N/A Timecode: Yes TDIR: Yes	PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz
.mxf Panasonic P2 (OP-Atom)	AVCIntra Class 50 1440x1080	Alpha: No Bitrate: 100 Bit depth: 10 VBI: N/A Timecode: N/A TDIR: Yes	No Audio
.mxf Panasonic P2 (OP-Atom)	AVCIntra Class 100 1920x1080	Alpha: No Bitrate: 100 Bit depth: 10 VBI: N/A Timecode: N/A TDIR: Yes	No Audio

MPEG-IFrame

Container	Codec	Features	Audio
.avi	MPEG2-IFrame422 1920x1080	Alpha: No Bitrate: 50-300 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

ProRes

Container	Codec	Features	Audio
.mov	I-Frame 4:2:2 1920x1080	Alpha: No Bitrate: 100 (LT), 147, 220 (HQ) Bit depth: 8, 10 VBI: N/A Timecode: N/A TDIR: Yes, with `.Ref` file	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

Uncompressed

Container	Codec	Features	Audio
.avi	Uncompressed YUYV 4:2:2 1920x1080	Alpha: No Bitrate: 2000 Bit depth: 8 VBI: N/A Timecode: N/A TDIR: Yes	No Audio PCM: - 2 ch: 24 in 32bit; 48kHz - 4 ch: 24 in 32bit; 48kHz - 8 ch: 24 in 32bit; 48kHz - 16 ch: 24 in 32bit; 48kHz

2160p50 and 2160p59.94 (60M) Codecs

XAVC

Container	Codec	Features	Audio
.mxf (OP1a)	MPEG2-IFrame422 3840x2160	Alpha: No Bitrate: 300, 480, (VBR/CBR) Bit depth: 10 VBI: N/A Timecode: Yes TDIR: N/A	PCM: - 16 ch: 24 in 32bit; 48kHz

3 Service Host component

Service Host is a Vizrt component that abstracts the Windows Service layer. The Service Host operates with plugins that implement a desired service or functionality. The Service Host itself takes care of logging and all the Windows Service stuff.

An example is the Control Mode plugin, which by default is registered as a Windows service called ServiceHost.control. Control Mode provides a REST interface and comes with a web-based user interface. It provides the necessary functionality to manage the Service Host plugins and installed Viz Engine instances.

3.1 Install / Uninstall

3.1.1 Installers

The Service Host installer comes in two flavors

- `ServiceHostBundle-x64*.exe` and
- `ServiceHost-x64*.msi`

The main difference is that the bundle installer takes care of upgrading an existing installation and installs the required operating system run-times if not already present whereas the msi installer just installs the Service Host.
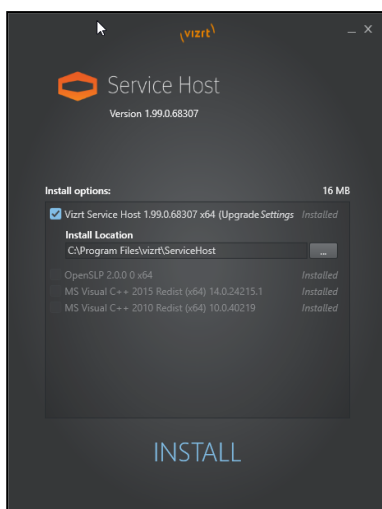
Installation

The bundle installer itself is using the msi installer. The msi and other files can be extracted from the bundle. This is mostly used for silent and unattended installs. Please run the bundle with -h command line switch to get all the available options.

Command Line Options for Bundle Installer

```
Z:\my_path> ServiceHostBundle-x64-1.0.0.68324 -h

Z:\my_path>
Running with no parameters will install the product
  -s, --silent            Runs the installer with no user interaction

     --msi                Extracts all .msi files to a subdirectory
     --dump               Extracts all files to a subdirectory
     --dumpTo=VALUE       Extracts all files to the specified path
  -h, -?, --help          Prints this help
```

msi Installer

```
Z:\my_path> msiexec /package ServiceHost-x64-1.99.0.68094.msi /l*vx! install.log
```

Pre Condition

- no Service Host is installed prior to the procedure

Post condition

- Service Host is installed at the indicated location. the default value is `%ProgramFiles%/vizrt/ServiceHost`
- configuration and log files are located at `%ProgramData%/vizrt/ServiceHost`
- if the configuration of the `Control Mode` does not exist, then a default one will be created at `%ProgramData%/vizrt/ServiceHost/ServiceHost.control.xml`
- the `Control Mode` is registered as Windows Service and started
- a shortcut, `ServiceHost.control`, is created in `%ProgramData%/vizrt/ServiceHost` to point to the web GUI of the `Control Mode`

Use this shortcut to verify the installation.

Upgrade an Existing Installation

msi Installer

Upgrade of an existing installation by using the msi installer is possible when doing an uninstall of the old package followed by an install procedure of the new package.

Pre Condition

- Service Host is installed
- the `Control Mode` is running

The uninstallation process write the current configuration and state into `Reinstall.profile`. This file is then used by the installer of the new package to re-establish the state and configuration.

Post Condition

- the new version of Service Host is installed
- the existing configuration and state of the Service Host services is re-established

Uninstall

msi Installer

```
Z:\my_path> msiexec /uninstall ServiceHost-x64-1.99.0.68094.msi /l*vx! uninstall.log
```

Pre Condition

- Service Host is installed
- zero, one or more services may be registered with this Service Host
- zero, one or more services may be registered with this Service Host and running
- the `Control Mode` has to be running. otherwise the uninstall silently fails

Post Condition

- The configuration and status of services under this Service Host are stored as the config file location in `Reinstall.profile`. They can be later used for further installations.
- any running services are stopped
- any registered services are unregistered
- the files under `%ProgramFiles%/vizrt/ServiceHost` are removed

Note

If the `Control Mode` is unreachable during uninstallation, it is still possible to uninstall. The dialog will inform users that the `Control Mode` is unreachable and prompt users to decide to force uninstall, or cancel. But if users decide to do force uninstall, users have do unregister all Service Host services registered as Windows services manually.
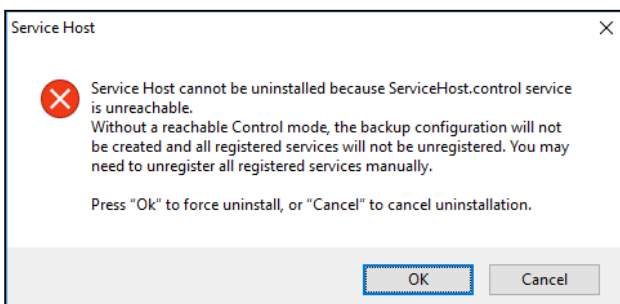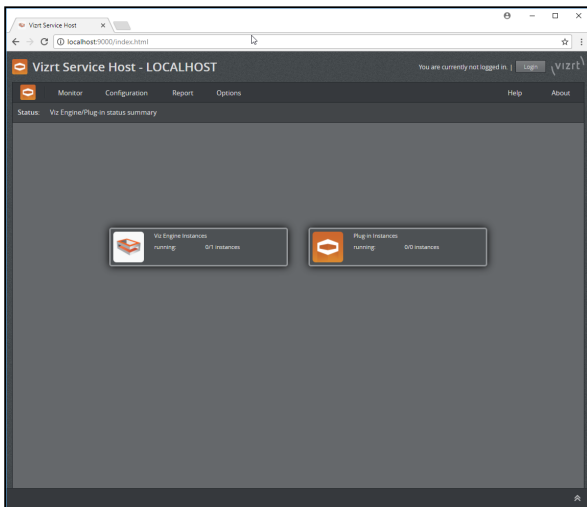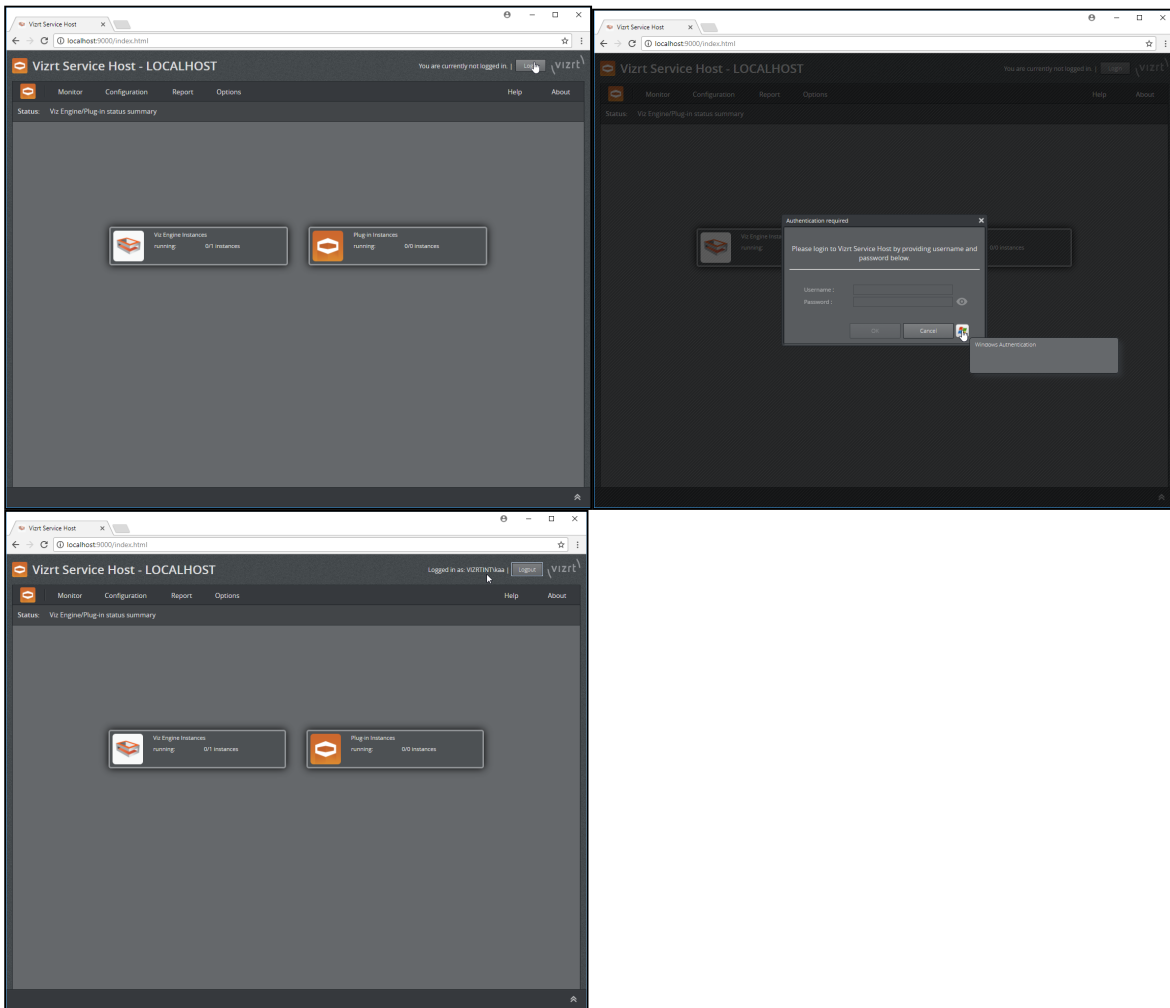
Control Mode

if one wants to register/unregister and start/stop Service Host plugin instances manually please take a look at Control Mode .

3.2 Control Mode

The `Control Mode` plugin is registered as a Windows Service when the Service Host is installed. This service is the administration point for this Service Host. The shortcut to the `Control Mode` of the current Service Host can be found at `%ProgramData%/vizrt/ServiceHost/ServiceHost.control`. It provides a REST interface to access the following functionalities.

On the welcome page, the first thing to do is to log in. The login name is used to log all interactions with the Control Mode. The login name can be viewed on the upper right of the GUI, next to the `Login/ Logout` button.

3.2.1 Viz Engine

The `Control Mode` is aware of Viz Engine instances on the local machine. They can be controlled from here.
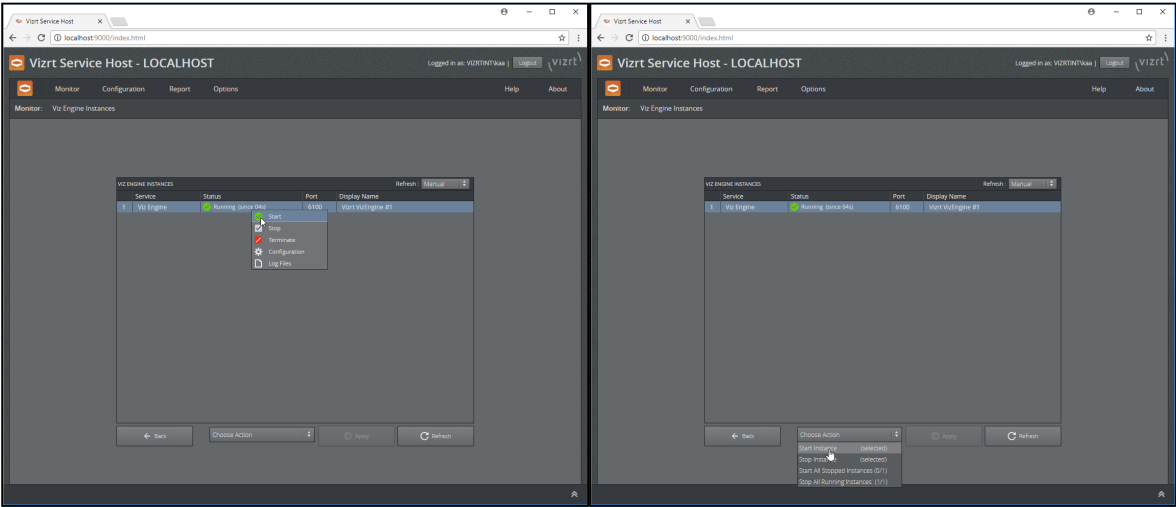
Follow the menu Monitor>Viz Engine
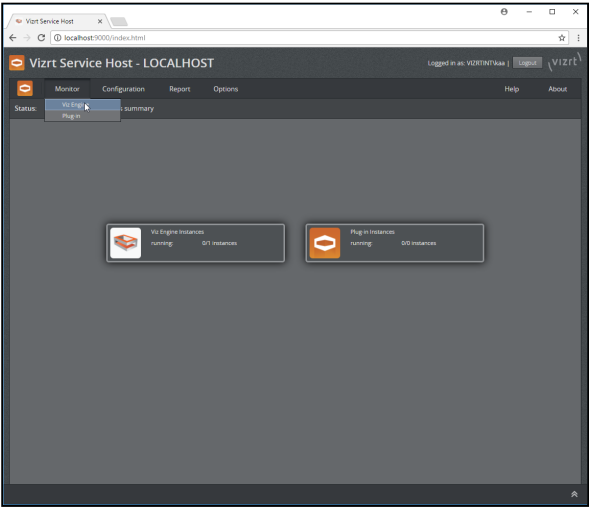
to go to this screen.
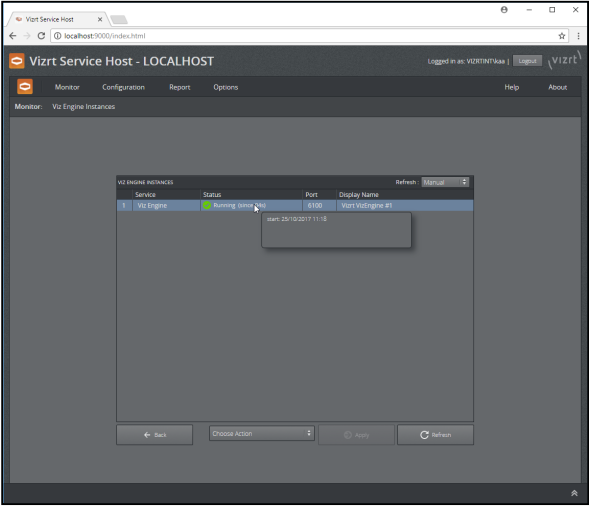
Start a Viz Engine instance

From the Viz Engine Instances Monitor page, one can use the context menu on an individual Viz Engine instance or use the action bar on the bottom of the screen to start Viz Engine instances.

The page is not automatically refreshed by default. This is due to performance reasons to keep the impact on the Viz Engines as minimal as possible. To check for a proper launch or terminate use the Refresh button or set a refresh rate.
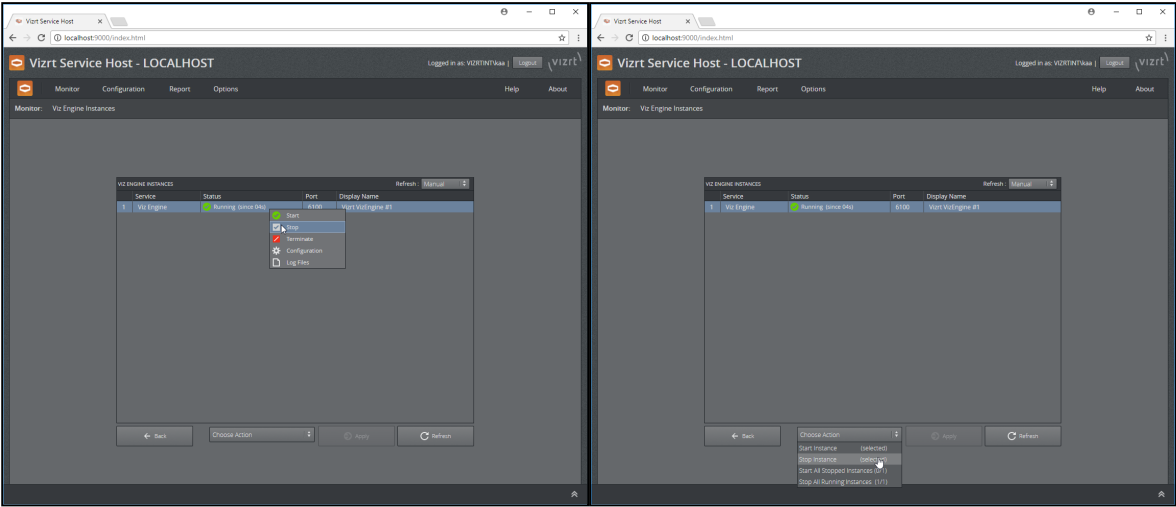
Precondition

- The Viz Engine needs to be installed at the default location

Postcondition

- start: a Viz Engine was launched without GUI and without console
- start: the launch may have failed. Please check the reason in the logs. Context menu Logs or Menu Report>Logs>Viz Engine

Stop a Viz Engine instance

From the Viz Engine Instances Monitor page one can use the context menu on an individual Viz Engine instance or use the action bar on the bottom of the screen to stop Viz Engine instances.

The page is not automatically refreshed by default. This is due to performance reasons to keep the impact on the Viz Engines as minimal as possible. To check for a proper launch or terminate use the Refresh button or set a refresh rate.

Precondition
- The Viz Engine needs to be installed at the default location
- The General Comm. Port of each Viz Engine instances need to be configured properly. Otherwise, the control instance will not be able to stop the specified Viz Engine instance correctly.

Postcondition
- stop: the Viz Engine instance was sent the EXIT command.

The shutdown of a Viz Engine may take some time. To kill the process, with all consequences, use Terminate from the context menu.

View Viz Engine Configuration

The config file of a Viz Engine instance can be viewed either from the Context menu on the Monitor page or via menu Configuration>Viz Engine
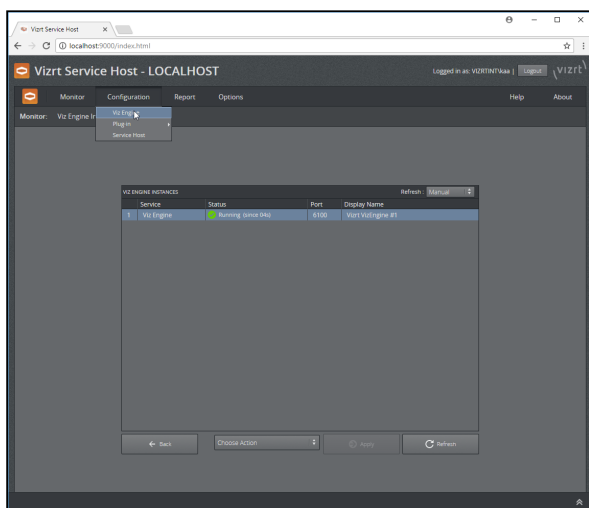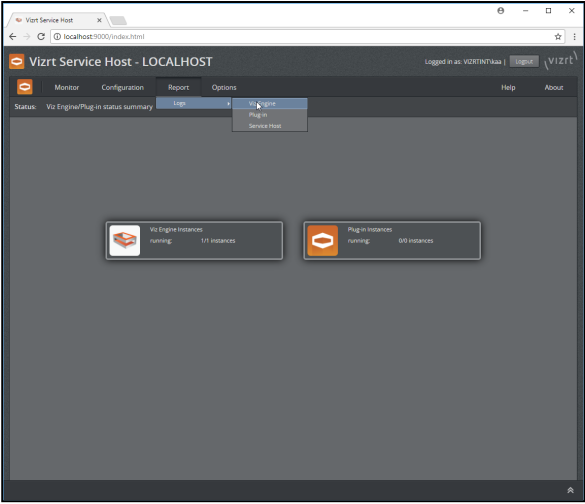

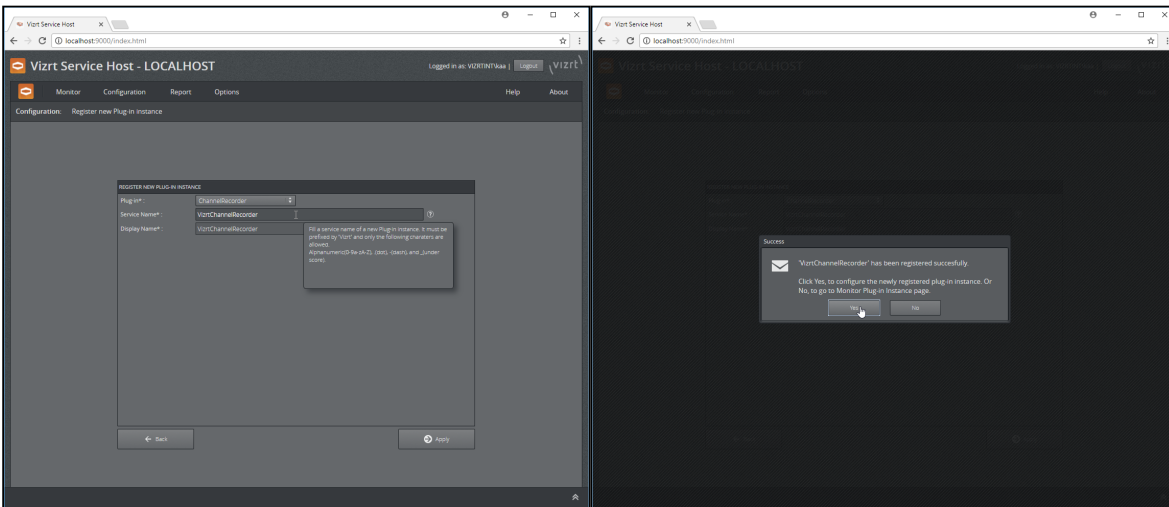From there one can select the Viz Engine instance
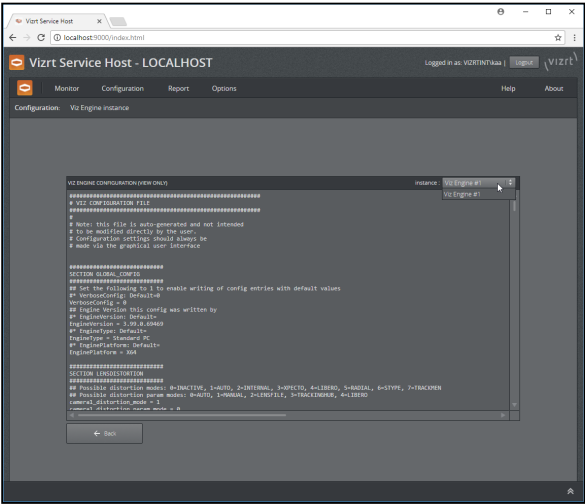
The page is view only.

View Viz Engine Logs

The log file of a Viz Engine instance can be viewed either from the Context menu on the Monitor page or via menu Report>Logs>Viz Engine.


From there one can select the Viz Engine instance to filter the available log files and how many lines to display.

3.2.2 Plugins

Register a Plugin Instance

To register a new instance of a plugin as a Windows Service, use the menu Configuration>Plugin-in>Register New Instance.

On the following screen one can select the plugin type and give the instance a service and display name. E.g. register a Channel Recorder:

Confirm with Apply. The dialog allows to go directly to the configuration of this plugin instance.
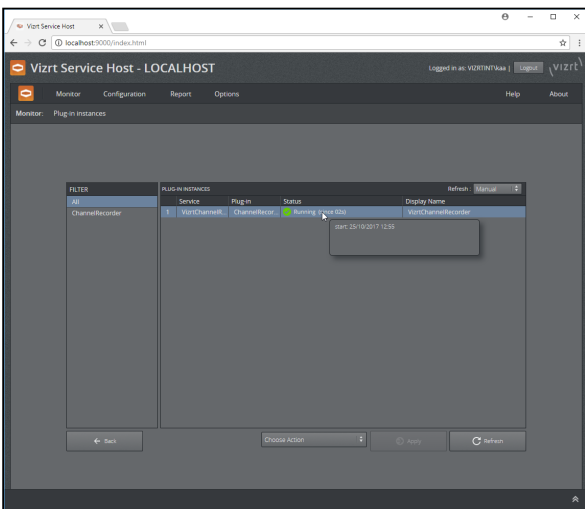
Configure a Plugin Instance

The configuration of a plugin instance can be reached from the Monitor plugin page via the context menu of a plugin instance or via the menu Configuration>Plug-in>Configure Instance.
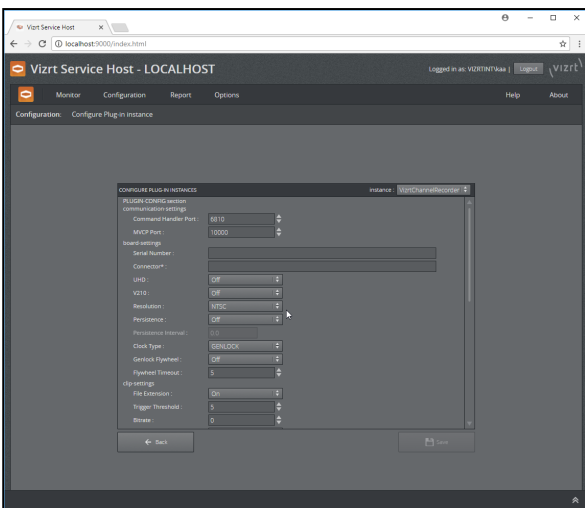
Plugin Monitor

The plugin monitor page can be reached via the menu Monitor>Plugin or from the welcome page by clicking on the Plugin-in Instances icon.

On this page we have a filter to the left hand side to select from the available plugin instances. Each concrete entry on the right hand side has a context menu that allows us to also unregister this plugin instance.

Unregister a Plugin Instance

The context menu provides the capability to unregister a plugin instance. Once selected and the following dialog has been answered with Yes then the plugin instance is stopped if it was running and then unregistered from the Windows services.

Any log and configuration files remain in ProgramData in order for later use by a subsequent new registration.
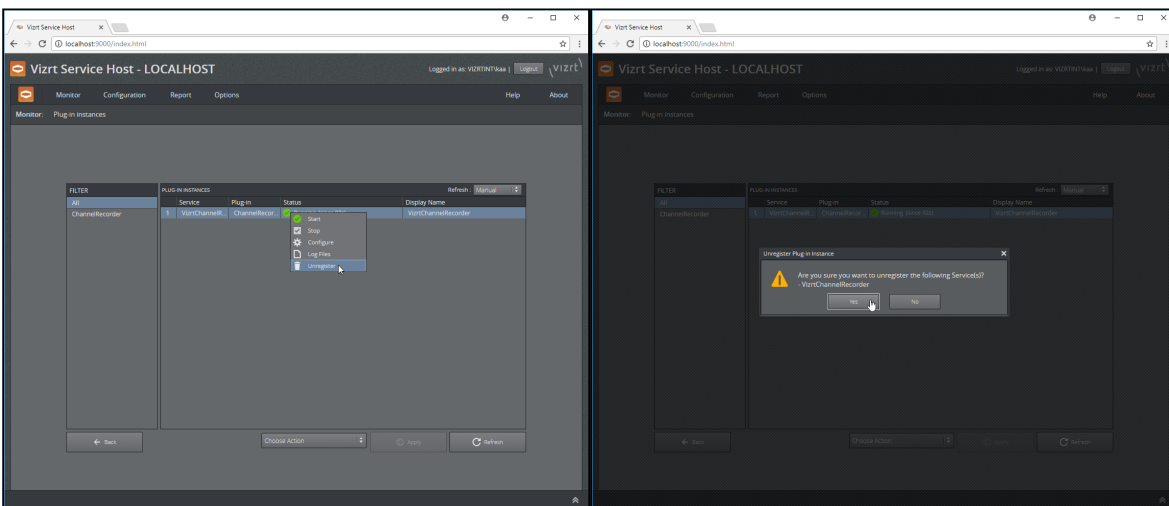
Report/Logs

The logs can be reached either via the context menu on the plugin monitor page or from the menu Report>Logs>Plug-in.
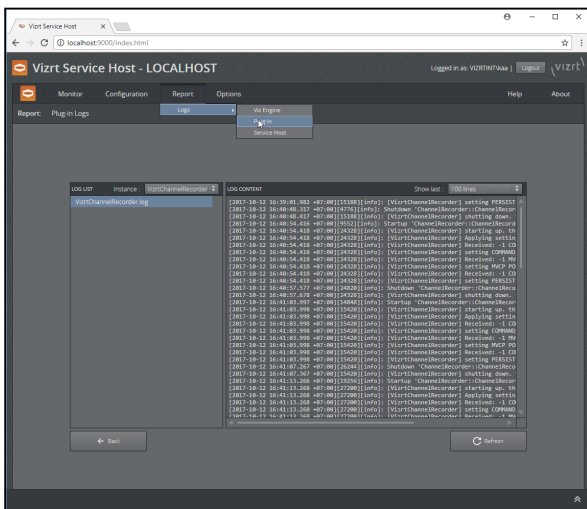
To the left there is the instance filter and the content of the selected log file is displayed on the right section of the screen. The number of lines displayed can be selected. It operates like the `tail` command.

3.2.3 Service Host (Control Mode)

The `Control Mode` itself can be configured too. For security reasons, the configuration of the `Control Mode` needs to be configured only by admins.

Configuration

The configuration page for the `Control Mode` may be reached via the menu Configuration>Service Host.
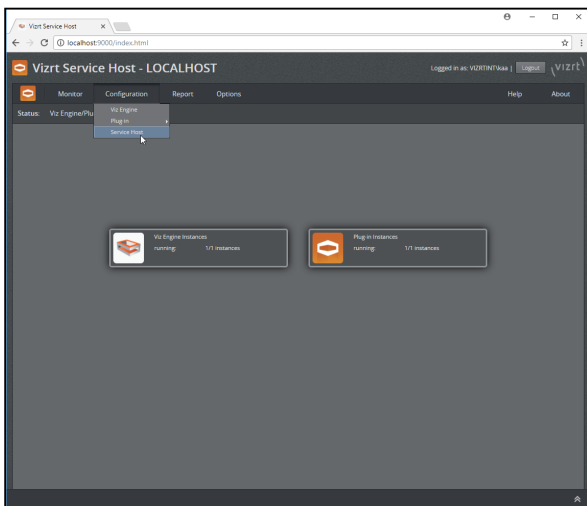
On this page the REST listening address can be configured. By default, Service Host is listening to all interfaces. This is specified by `[All interfaces]` in the `Listen to` field. As port number, any valid and unused port number may be used. Later versions may support automatically finding of an available port number.

The Access Treat Type allows to control who is able to interact with the Service Host.

- Only local access(`LOCAL_ONLY`)

To allow only users accessing the GUI via the local machine to change the sysyem(such register/ unregister new plugin instance, or start/stop Viz Engine instances). For users accessing the GUI from external machine whether authenticated or not are not allowed to do any changes.

- Authentication required(`AUTHENTICATION_REQUIRED`)

Same as above, Only local access, except that users accessing from external are allowed as well as long as they are authenticated. In future versions this may change to all users are required to authenticate before they can make changes to the system.

- No authentication(`NO_AUTHENTICATION`)

No restrictions apply. All users may change the system, whether local or from external. We do no recommend this setting since there is also no way of having a user information in the operating logs.
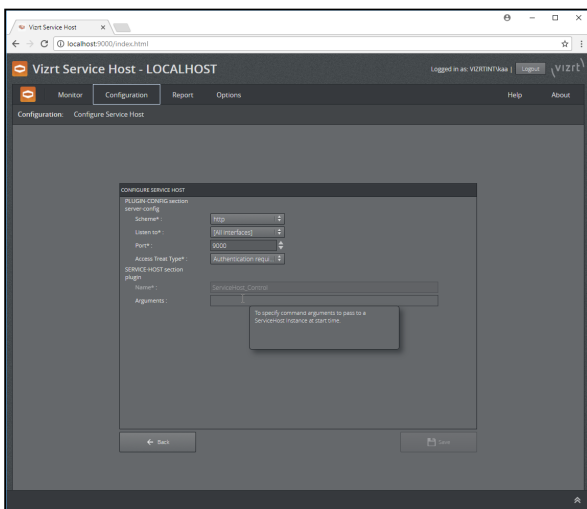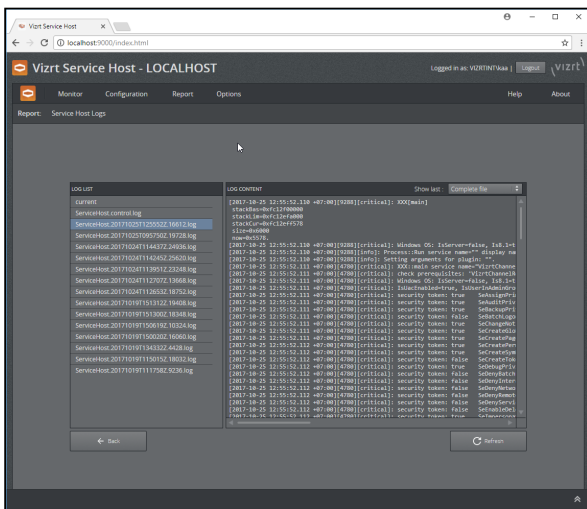

The Arguments edit lets one pass arbitrary startup arguments to the control mode plugin. In this case we set the log level to debug mode.

Allowed values for Arguments are

- -l debug
- -l info
- -l warning
- -l error
- -l off

Report/Logs

The logs produced by the control mode may be accessed via the menu Report>Logs>Service Host.

On the left hand side one can select the log file to be displayed. There are two types of logs to be found.

The Control Mode log

This log file is named `ServiceHost.control.log` It does log rotation and may also be reached via the log panel on the bottom of the screen.

 and

Service Host Startup Logs

This is produced during startup and kept as long as this Service Host is running. That way one can capture possible startup issues before a plugin is loaded and initialized. These log files are removed after a successful shutdown. They are named `ServiceHost.dateandtime.pid.log` where `dateandtime` is the date and time in Z timezone, `pid` is the process id.

3.2.4 Debugging of Service Host

Manual Register/Unregister and Stop/Start

The Service Host itself is capable to register/unregister itself as `Control Mode` service. This might be useful in cases not covered by the installer.

Register Control Mode from Command Line

```
Z:\> ServiceHost.exe -c -i
```

Unregister Control Mode from Command Line

```
Z:\> ServiceHost.exe -c -r
```

The Service Host in `Control Mode` can also be started and stopped from the command line.

Start Control Mode from Command Line

```
Z:\> ServiceHost.exe -c -s
```

Stop Control Mode from Command Line

```
Z:\> ServiceHost.exe -c -k
```

Launch Service Host in Foreground With a Console Window

For debugging the Service Host can be launched in foreground and with a console window attached. To list all the possibilities use the command line switch -h

---

Command line Options for Service Host

```
Z:\>ServiceHost.exe -h
Usage:
$0 [-v] ... start from within windows services, optional verbose mode
$0 [-v] -N service_name -i
      ... install $service_name as service
          -v ... optional verbose mode
$0 [-v] -N service_name -r
      ... remove $service_name as service
          -v ... optional verbose mode
$0 [-v] -N service_name -s
      ... start service $service_name
          -v ... optional verbose mode
$0 [-v] -N service_name -k
      ... kill service $service_name
          -v ... optional verbose mode
$0 [-v] -N service_name -d
      ... run $service_name in debug mode with a console window
          -v ... optional verbose mode
$0 [-v] -c -i
      ... install control as service, the service name is '{ProductName}.control'
          -v ... optional verbose mode
$0 [-v] -c -r
      ... remove control as service, the service name is '{ProductName}.control'
          -v ... optional verbose mode
$0 [-v] -c -s
      ... start control service, the service name is '{ProductName}.control'
          -v ... optional verbose mode
$0 [-v] -c -k
      ... kill control service, the service name is '{ProductName}.control'
          -v ... optional verbose mode
$0 [-v] -c -d
      ... run control in debug mode with a console window
          -v ... optional verbose mode
-p "[argument...]"
      ... arguments that gets forwarded to the plugin
          e.g. -p "-l debug -v"
```

What we are interested in here are the lines with the -d option.

To start the Service Host in the `Control Mode` with a console window make sure it is not running as windows service then launch it with -c -d options.

```
Z:\>ServiceHost.exe -c -k
Z:\>ServiceHost.exe -c -d
```

optionally add verbose and log parameters to the Service Host and the `Control Mode` plugin.

```
Z:\>ServiceHost.exe -c -d -v -l debug -p "-v -l debug"
```

the first `-v -l debug` are for the Service Host executable and the one inside the `-p` option are forwarded to the `Control Mode` plugin. In the console mode, the logs are also forwarded to the console window.