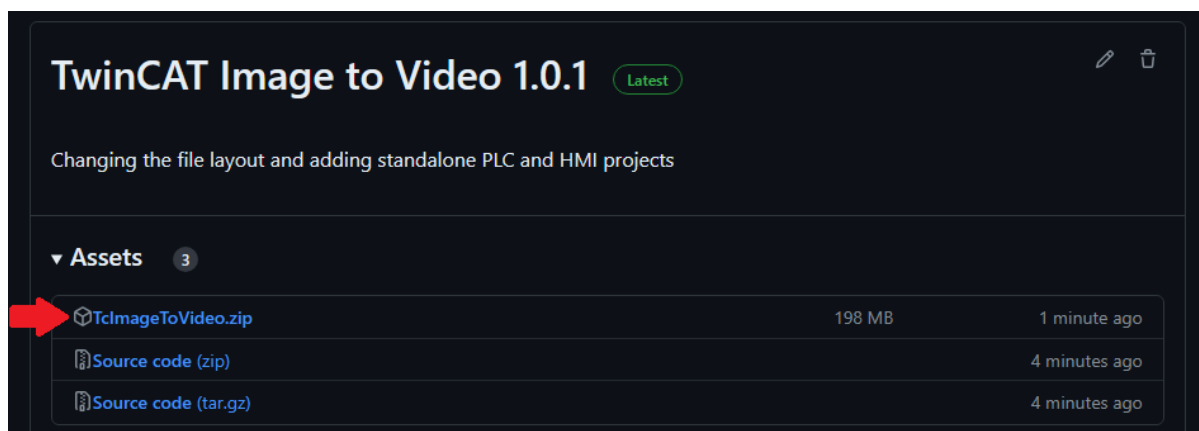


Event Video Playback

This repository includes both the source files and the release package for the TwinCAT Event Video Playback package. The package provides an easy to use PLC interface for assembling images captured with TwinCAT Vision into a single video file. When the video file is created, a corresponding alarm event is logged into the TwinCAT Event Logger for later viewing. In addition, an HMI Control component is supplied for easy viewing and playback of logged video events on TwinCAT HMI.

Getting Started

Instead of cloning the source repository for this project, start with the installer zip files located in the [Releases section](#) of this repository. Download the latest copy of the TcImageToVideo.zip from the Releases section. The release package will include a sample PLC project, sample HMI project, PLC library, and the required Windows service installer.



First Steps!!! - Install Instructions

All these file necessary are included inside the TcImageToVideo.zip under the [Install Files](#).

TcVision .Net Service

1. Install .Net Runtime 8.0.4 for Windows x64 (**dotnet-runtime-8.0.4-win-x64.exe**)
2. Install the TwinCAT Image to Video Service (**ImageToVideoSetup.msi**)

PLC Library

1. Either open the included Quick Start Sample, or a new PLC Project
2. Right Click References under PLC Project -> **Library Repository**
3. Click the **Install** button, and install the included SPT Vision library

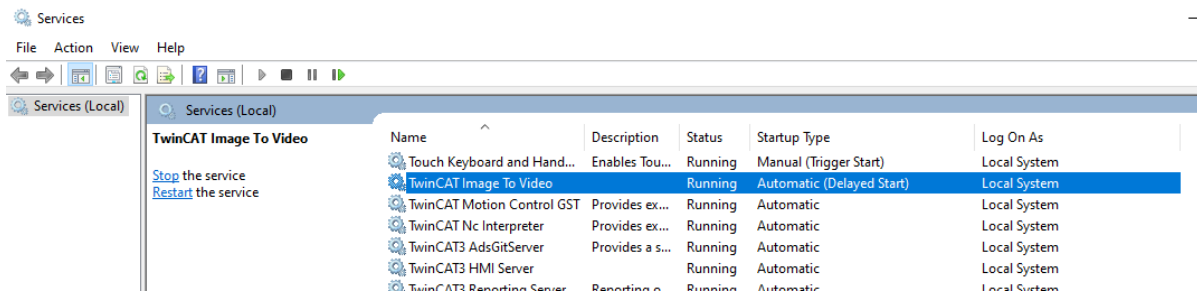
TwinCAT HMI

1. Copy the **EventVision.'version'.nupkg** file to the directory **C:\TwinCAT\Functions\TE2000-HMI-Engineering\References**.

TwinCAT Image to Video Service

To use the Event Video Playback feature, you must first install the ImageToVideo service. This service will run in the background of the IPC and convert TwinCAT Vision images to video files when called from the PLC.

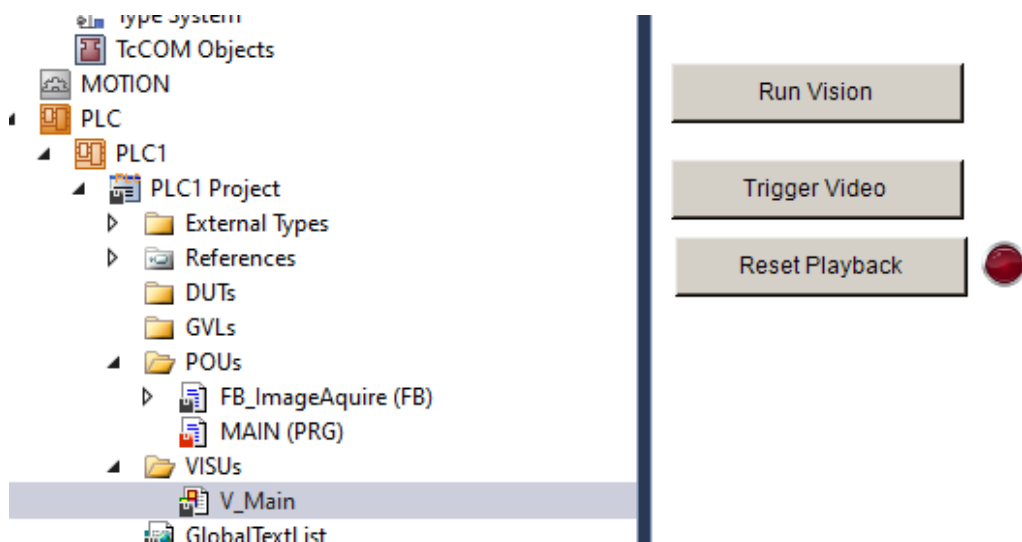
Install the service via the supplied MSI installer file, and you can verify with the Windows Services.



PLC Project - Quick Start Sample

For quick start purposes, a PLC sample project and the SPT_Vision library is supplied. To run the sample, follow these steps.

1. To open the .tnzip project files go to File -> Open -> Open From Archive
2. Open PLC project and install the SPT_Vision library
3. Activate the TwinCAT project
4. Load the included sample images into the TC Vision File Source
5. Put the PLC into Run state
6. Open the Visu project and press the Run Vision button to start the vision process



7. Check images are streaming via the TwinCAT -> Windows-> ADS Image Watch
8. To generate a video, press the Trigger Video button on the Visu
9. Check that a video was created in the default directory C:\Videos

Notes on PLC Project

- The SPT_Vision component FB_ImageAquire and the base Tc3 Vision components are supported. For the sample, the SPT_Vision component FB_ImageAquire is used. The FB_ImageToVideo function does not require FB_ImageAquire to work, and it can also be supplied with standard image pointers from the Tc3 Vision base library.

- The FB_ImageToVideo is configurable. You can set the directory of where the video will be recorded, as well as the Camera name and Event Logger name.

```
Playback : FB_ImageToVideo := (CameraName      := 'Camera1',
                               JsonAttribute   := '{CreateVid : 1, CameraName: "Camera1"}',
                               FramesPerSecond := 10,
                               TimeBeforeEvent := T#7S,
                               TimeAfterEvent  := T#3S,
                               VideoOutputDirectory := 'C:\Videos',
                               ReductionFactor  := 0.25,
                               Codec           := 'H264');
```

Note

If you are not able to play the video files on your system due to codec warning, try to adjust the codec settings; "Default", "AVC", "H264". You should be able to simply double click the video file and play it in Windows Media Player.

PLC Project - Adapting to Existing Vision Projects

You can easily make use of this new service with existing TcVision Projects. There are 4 main components that need to be addressed.

1. Add the SPT Vision Library to the References section of the PLC Project (v 3.0.2.1 or later)
2. Instantiate a new FB_ImageToVideo and TriggerEvent BOOL

```
// ImageToVideo Instance
Playback : FB_ImageToVideo := (CameraName      := 'Camera1',
                               JsonAttribute   := '{CreateVid : 1, CameraName:
"Camera1"}',
                               FramesPerSecond := 10,
                               TimeBeforeEvent := T#7S,
                               TimeAfterEvent  := T#3S,
                               VideoOutputDirectory := 'C:\Videos',
                               ReductionFactor  := 0.25,
                               Codec           := 'H264');

// Event Trigger Boolean
TriggerEvent : BOOL;
```

3. Add the CyclicLogic call to the main body of your POU. This **MUST** be called cyclically to work.

```
Playback.CyclicLogic();
```

4. Add the trigger logic somewhere in your program. The TriggerAlarmForVideoCapture method only needs to be called once to start processing.

```
IF TriggerEvent THEN
    TriggerEvent := FALSE;
    Playback.TriggerAlarmForVideoCapture();
END_IF
```

5. Add the AddImage method to your image aquisition loop of your program. This will add an image to the buffer of the Playback block.

```
Playback.AddImage(ipImageIn := ImageIn);
```

HMI Control Component

Included in the package is a NuGet Package for the EventVision component. To install, copy the EventVision.'version'.nupkg file to the directory C:\TwinCAT\Functions\TE2000-HMI-Engineering\References.

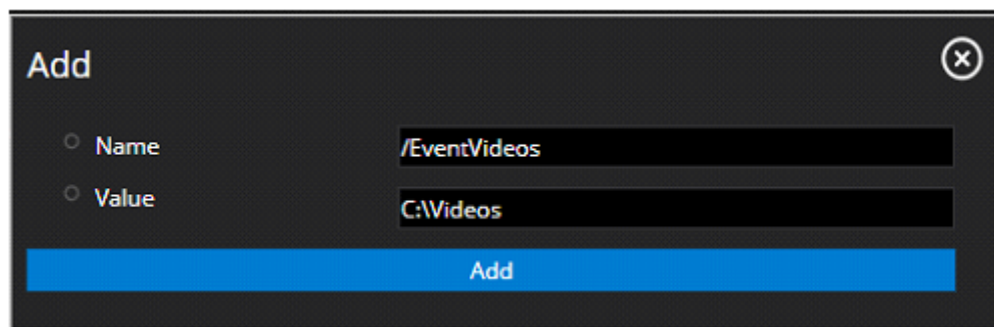
To configure the project:

1. Install the **Beckhoff.TwinCAT.HMI.EventLogger** package with the NuGet Package Manager
2. On the Browse window of the NuGet Package Manager, install the **EventVision** package

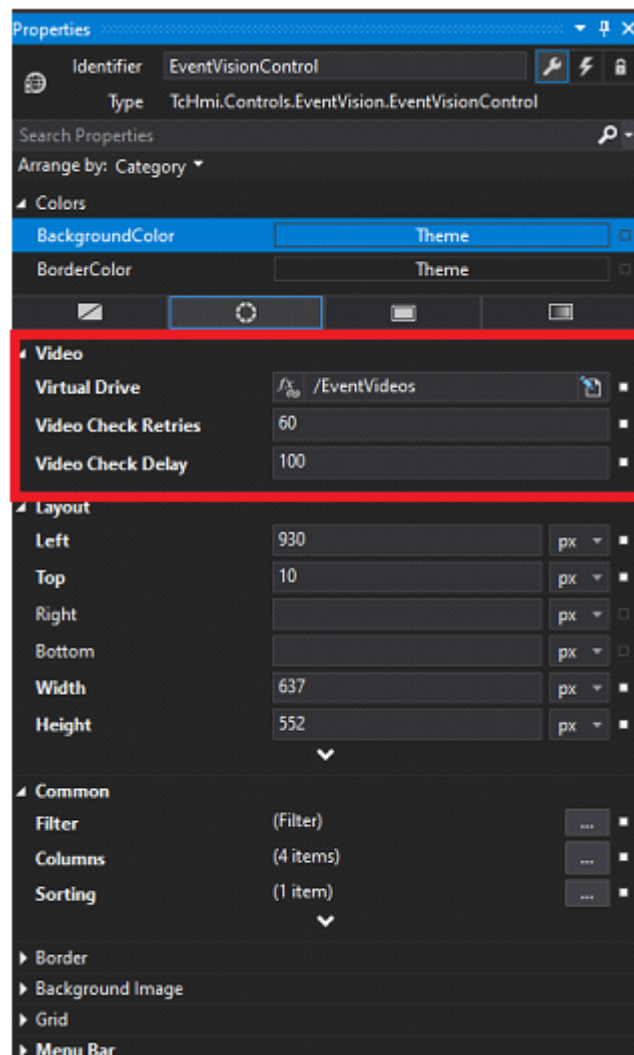
Tip

If the package does not appear, double check Package source has TwinCAT HMI Customer profile selected

3. Set a virtual directory of where the video files exist. Go to Server -> TcHmiSrv -> Virtual directories and add the following entry.



4. On your HMI view, add a new EventVideo Control grid
5. Modify the top 3 properties and make sure the Virtual Drive matched the Virtual Directory created in Step 3



6. Launch the HMI in a browser

Important

The EventVideoPlayback control will only work in browser launch or fully deployed environment. HMI Live View will not allow you to play video files due to the virtual directory usage.

7. Double click the Video event to play

Details

Domain	TcHmiEventLogger
Source domain	TcHmiEventLogger
Text	Camera1 Alarm
Name	TCEVENTLOGGER_ALARM
Type	Alarm
Severity	Error
Alarm state	Cleared, no confirmation needed
Raised	5/2/2024, 11:56:40.983 AM
Cleared	5/2/2024, 11:56:47.974 AM
Confirmed	-
Parameters	Data: Q2FtZXJhMQA= Event Class: {A53F9540-5C3E-4D26-A197-D1116085697F} Event Class Name: EventClass1_DisplayText Event Id: 1 Json Attribute: {CreateVid: 1, CameraName: "Camera1"} Localized Source Name: EventText not found! ==> EventId = 1530539511 Num Arguments: 1 Source Id: 139468800 Source Name: MAIN.Playback Target Name: Local Types: DgA=

PLAY VIDEO

Close

Note

- If you only see a blank playback window, try adjusting the codec that was used to record the video
- If you get a "Waiting For Video" in place of "Play Video", try the following:
 - Re-adding the EventVideo Control
 - Completely remove the virtual directory and re-add
 - Make sure Check Retires and Check Delay is set in the properties window of the control

Dependencies

C# Service (only if building from source):

- Visual Studio Pro 2022
- Microsoft Visual Studio Installer Projects 2022 Extension

PLC:

- TwinCAT 3 build 4024 or higher
- TF7000
- SPT_Vision Library
- ImageToVideo Service

HMI:

- Beckhoff.TwinCAT.HMI.EventLogger NuGet Package

- EventVideo NuGet Package

How to get support

Should you have any questions regarding the provided sample code, please contact your local Beckhoff support team. Contact information can be found on the official Beckhoff website at <https://www.beckhoff.com/en-us/support/>.

Disclaimer

All sample code provided by Beckhoff Automation LLC are for illustrative purposes only and are provided “as is” and without any warranties, express or implied. Actual implementations in applications will vary significantly. Beckhoff Automation LLC shall have no liability for, and does not waive any rights in relation to, any code samples that it provides or the use of such code samples for any purpose.