---

layout: doc title: HMI NuGet Package Usage description: Integrate video playback controls into your TwinCAT HMI and WPF applications permalink: /docs/hmi-usage/

Overview

The Event Video Playback HMI NuGet package provides ready-to-use WPF controls for displaying and controlling video playback in your HMI applications. This guide covers installation, configuration, and usage.

Installation

Prerequisites

- **Visual Studio 2019 or later**
- **TwinCAT HMI Framework** (for HMI projects) or standalone **WPF application**
- **.NET 8 Framework**
- **NuGet Package Manager**

Install via NuGet

Option 1: Package Manager Console

```
Install-Package EventVideoPlayback.HMI
```

Option 2: NuGet Package Manager UI

1. Right-click on your project in Solution Explorer
2. Select **Manage NuGet Packages**
3. Click on **Browse** tab
4. Search for "EventVideoPlayback.HMI"
5. Click **Install**

Option 3: Beckhoff USA Community Package Feed

Add the Beckhoff USA Community package feed to your NuGet sources:

1. **Tools** > **Options** > **NuGet Package Manager** > **Package Sources**
2. Click the **+** button
3. Name: `Beckhoff USA Community`
4. Source: `https://packages.beckhoff-usa-community.com/nuget/v3/index.json`
5. Click **OK**

Then search for and install the package.

Video Player Control

Basic Usage

Add the video player control to your XAML:

```
<Window x:Class="YourNamespace.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:evp="clr-namespace:EventVideoPlayback.HMI.Controls;assembly=EventVideoPlayback.HMI"
        Title="Event Video Playback" Height="600" Width="800">

    <Grid>
        <evp:EventVideoPlayer x:Name="videoPlayer"
                              VideoSource="{Binding CurrentVideoPath}"
                              AutoPlay="True"
                              ShowControls="True"
                              Volume="0.5" />
    </Grid>
</Window>
```

Control Properties

Property	Type	Default	Description
`VideoSource`	string	null	Path to the video file to play
`AutoPlay`	bool	false	Automatically start playback when source is set
`ShowControls`	bool	true	Display playback controls
`Volume`	double	0.5	Volume level (0.0 to 1.0)
`IsMuted`	bool	false	Mute audio
`Loop`	bool	false	Loop video playback
`Stretch`	Stretch	Uniform	How video is stretched to fill control

Control Methods

```
// Play the video
videoPlayer.Play();
```

```
// Pause the video
videoPlayer.Pause();

// Stop the video
videoPlayer.Stop();

// Seek to specific position (in seconds)
videoPlayer.Seek(10.5);

// Load a new video
videoPlayer.LoadVideo("C:\\Videos\\MachineEvent_001.mp4");

// Take a snapshot
videoPlayer.SaveSnapshot("C:\\Snapshots\\snapshot.png");
```

Events

```
// Video loaded and ready to play
videoPlayer.VideoLoaded += (sender, e) =>
{
    Debug.WriteLine($"Video loaded: {e.VideoPath}");
};

// Playback started
videoPlayer.PlaybackStarted += (sender, e) =>
{
    Debug.WriteLine("Playback started");
};

// Playback paused
videoPlayer.PlaybackPaused += (sender, e) =>
{
    Debug.WriteLine("Playback paused");
};

// Playback completed
videoPlayer.PlaybackCompleted += (sender, e) =>
{
    Debug.WriteLine("Playback completed");
};

// Error occurred
videoPlayer.ErrorOccurred += (sender, e) =>
{
    MessageBox.Show($"Error: {e.ErrorMessage}", "Video Error");
};
```

Complete Example Application

XAML (MainWindow.xaml)

```
<Window x:Class="VideoPlaybackDemo.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:evp="clr-namespace:EventVideoPlayback.HMI.Controls;assembly=EventVideoPlayback.HMI"
        Title="Event Video Playback Demo"
        Height="700" Width="1000"
        Loaded="Window_Loaded">

    <Grid Background="#1a1a1a">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*"/>
            <RowDefinition Height="Auto"/>
        </Grid.RowDefinitions>

        <!-- Header -->
        <Border Grid.Row="0" Background="#D32F2F" Padding="15">
            <TextBlock Text="Event Video Playback Viewer"
                       FontSize="24"
                       FontWeight="Bold"
                       Foreground="White"/>
        </Border>

        <!-- Video Player -->
        <evp:EventVideoPlayer Grid.Row="1"
                              x:Name="videoPlayer"
                              Margin="10"
                              ShowControls="True"
                              Volume="0.7"/>

        <!-- Video List and Controls -->
        <Grid Grid.Row="2" Margin="10">
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="*"/>
                <ColumnDefinition Width="Auto"/>
            </Grid.ColumnDefinitions>

            <!-- Video List -->
            <ListBox Grid.Column="0"
                     x:Name="videoListBox"
                     SelectionChanged="VideoListBox_SelectionChanged"
                     Background="#2a2a2a"
                     Foreground="White"
                     Padding="5"
                     Height="150">
                <ListBox.ItemTemplate>
                    <DataTemplate>
                        <StackPanel Orientation="Horizontal">
                            <TextBlock Text="■" Margin="0,0,10,0"/>
                            <TextBlock Text="{Binding Name}"/>
```

```
                        </StackPanel>
                    </DataTemplate>
                </ListBox.ItemTemplate>
            </ListBox>

            <!-- Control Buttons -->
            <StackPanel Grid.Column="1" Margin="10,0,0,0" VerticalAlignment="Center">
                <Button Content="Refresh List"
                        Click="RefreshList_Click"
                        Padding="15,5"
                        Margin="0,0,0,5"
                        Background="#D32F2F"
                        Foreground="White"
                        BorderThickness="0"
                        Cursor="Hand"/>
                <Button Content="Open Folder"
                        Click="OpenFolder_Click"
                        Padding="15,5"
                        Margin="0,0,0,5"
                        Background="#424242"
                        Foreground="White"
                        BorderThickness="0"
                        Cursor="Hand"/>
                <Button Content="Take Snapshot"
                        Click="TakeSnapshot_Click"
                        Padding="15,5"
                        Background="#424242"
                        Foreground="White"
                        BorderThickness="0"
                        Cursor="Hand"/>
            </StackPanel>
        </Grid>
    </Grid>
</Window>
```

Code-Behind (MainWindow.xaml.cs)

```
using System;
using System.IO;
using System.Linq;
using System.Windows;
using System.Diagnostics;
using EventVideoPlayback.HMI.Controls;

namespace VideoPlaybackDemo
{
    public partial class MainWindow : Window
    {
        private const string VIDEO_FOLDER = @"C:\TwinCAT\EventVideoPlayback\Videos";

        public MainWindow()
```

```
        {
            InitializeComponent();
        }

        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            // Subscribe to video player events
            videoPlayer.VideoLoaded += VideoPlayer_VideoLoaded;
            videoPlayer.ErrorOccurred += VideoPlayer_ErrorOccurred;
            videoPlayer.PlaybackCompleted += VideoPlayer_PlaybackCompleted;

            // Load initial video list
            RefreshVideoList();
        }

        private void RefreshVideoList()
        {
            try
            {
                if (!Directory.Exists(VIDEO_FOLDER))
                {
                    MessageBox.Show($"Video folder not found: {VIDEO_FOLDER}",
                                    "Error",
                                    MessageBoxButton.OK,
                                    MessageBoxImage.Error);
                    return;
                }

                var videoFiles = Directory.GetFiles(VIDEO_FOLDER, "*.mp4")
                                          .Select(f => new FileInfo(f))
                                          .OrderByDescending(f => f.CreationTime)
                                          .ToList();

                videoListBox.ItemsSource = videoFiles;

                // Auto-select first video
                if (videoFiles.Any())
                {
                    videoListBox.SelectedIndex = 0;
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error loading video list: {ex.Message}",
                                "Error",
                                MessageBoxButton.OK,
                                MessageBoxImage.Error);
            }
        }

        private void VideoListBox_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e)
        {
```

```
            if (videoListBox.SelectedItem is FileInfo fileInfo)
            {
                videoPlayer.LoadVideo(fileInfo.FullName);
            }
        }

        private void RefreshList_Click(object sender, RoutedEventArgs e)
        {
            RefreshVideoList();
        }

        private void OpenFolder_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                if (Directory.Exists(VIDEO_FOLDER))
                {
                    Process.Start("explorer.exe", VIDEO_FOLDER);
                }
                else
                {
                    MessageBox.Show($"Folder not found: {VIDEO_FOLDER}",
                                    "Error",
                                    MessageBoxButton.OK,
                                    MessageBoxImage.Warning);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error opening folder: {ex.Message}",
                                "Error",
                                MessageBoxButton.OK,
                                MessageBoxImage.Error);
            }
        }

        private void TakeSnapshot_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                string snapshotPath = Path.Combine(
                    VIDEO_FOLDER,
                    $"Snapshot_{DateTime.Now:yyyyMMdd_HHmmss}.png"
                );

                videoPlayer.SaveSnapshot(snapshotPath);

                MessageBox.Show($"Snapshot saved to:\n{snapshotPath}",
                                "Snapshot Saved",
                                MessageBoxButton.OK,
                                MessageBoxImage.Information);
            }
```

```
            catch (Exception ex)
            {
                MessageBox.Show($"Error saving snapshot: {ex.Message}",
                                "Error",
                                MessageBoxButton.OK,
                                MessageBoxImage.Error);
            }
        }

        private void VideoPlayer_VideoLoaded(object sender, VideoLoadedEventArgs e)
        {
            // Optional: Update UI when video loads
            Title = $"Event Video Playback - {Path.GetFileName(e.VideoPath)}";
        }

        private void VideoPlayer_ErrorOccurred(object sender, VideoErrorEventArgs e)
        {
            MessageBox.Show($"Video playback error:\n{e.ErrorMessage}",
                            "Playback Error",
                            MessageBoxButton.OK,
                            MessageBoxImage.Error);
        }

        private void VideoPlayer_PlaybackCompleted(object sender, EventArgs e)
        {
            // Optional: Auto-play next video
            if (videoListBox.SelectedIndex < videoListBox.Items.Count - 1)
            {
                videoListBox.SelectedIndex++;
            }
        }
    }
}
```

Integration with TwinCAT HMI

For TwinCAT HMI projects, the control can be integrated with symbol bindings:

```
<evp:EventVideoPlayer VideoSource="{Binding PLC.VideoPath, Mode=OneWay}"
                      AutoPlay="{Binding PLC.AutoPlayEnabled, Mode=OneWay}"
                      Volume="{Binding HMI.VolumeLevel, Mode=TwoWay}"/>
```

Styling and Customization

Custom Control Template

You can customize the appearance of the video player:

```
<evp:EventVideoPlayer x:Name="videoPlayer">
   <evp:EventVideoPlayer.Style>
       <Style TargetType="evp:EventVideoPlayer">
           <Setter Property="Background" Value="#1a1a1a"/>
           <Setter Property="Foreground" Value="White"/>
           <Setter Property="BorderBrush" Value="#D32F2F"/>
           <Setter Property="BorderThickness" Value="2"/>
       </Style>
   </evp:EventVideoPlayer.Style>
</evp:EventVideoPlayer>
```

Performance Tips

1. **Preload Videos**: Load videos in background threads to avoid UI freezing
2. **Hardware Acceleration**: Ensure GPU acceleration is enabled
3. **File Formats**: Use H.264 (avc1) codec for best compatibility
4. **Resolution**: Match video resolution to display size
5. **Memory Management**: Dispose of video player when not in use

Troubleshooting

Video Won't Play

- Verify the video file exists and is accessible
- Check that the codec is supported (H.264 recommended)
- Ensure .NET 8 runtime is installed
- Try playing the video in Windows Media Player to verify it's valid

Control Not Visible

- Check that the NuGet package is correctly installed
- Verify XAML namespace declaration
- Ensure the control has proper Width/Height or is in a sized container

Performance Issues

- Reduce video resolution
- Use hardware acceleration
- Close other resource-intensive applications
- Check CPU and GPU usage

Best Practices

1. **Error Handling**: Always subscribe to ErrorOccurred event
2. **Resource Cleanup**: Dispose of video player when closing window
3. **File Validation**: Verify video files exist before loading
4. **User Feedback**: Show loading indicators for long operations
5. **Testing**: Test with various video formats and resolutions

Next Steps

- Review PLC Library Usage to trigger video recordings
- Configure Service Settings for optimal performance
- Check Getting Started for installation help